CONTINUAL LEARNING IN AGENTIC RL VIA PROJECTED SUCCESS GRADIENT CORRECTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Agentic reinforcement learning can forget previously successful behaviors even within a single training run on a fixed task distribution. We quantify this trajectory-level forgetting, revealing a continual learning problem arising without an explicit sequence of tasks. At the trajectory level, maximum-return oracle trajectories have nonnegative advantage relative to the current policy's expected return, regardless of which policy generated them. This provides a policy-improvement motivation for selecting historical successes as protection targets. We introduce Success-Constrained Policy Gradient Projection, which makes the minimum Euclidean correction to the main loss gradient needed to satisfy an aggregate success-replay constraint. Our analysis characterizes the tradeoff between protecting historical successes and optimizing the current objective: for a gradient-descent step, the correction removes a first-order increase in aggregate replay loss, with a cost to the main objective determined by gradient alignment. Averaged over three training seeds on WebShop, the method improves success by 6.2 and 2.1 percentage points for GRPO and GiGPO, respectively, without changing their advantage estimators. GRPO with the correction reaches 71.4% mean success, surpassing the GiGPO baseline at 70.2%. In a supplementary ALFWorld comparison using one training seed without policy collapse, the method improves final success and reduces the decline from validation-selected saved checkpoints with both algorithms. These results demonstrate that retaining historical successes can yield gains comparable to finer-grained advantage estimation, and establish the practical value of success-based protection for continual learning in agentic reinforcement learning.

1 INTRODUCTION

Group-based RL methods such as GRPO and GiGPO train LLM agents on trajectories of multi-turn environment interaction (Shao et al., 2024; Feng et al., 2025a). An update that improves the current training objective can nevertheless lower the likelihood of a trajectory that completed a task earlier in training. This raises a retention question within a single RL run: can policy optimization preserve successful behavior while learning from current rollouts? The task distribution need not change for this problem to arise, because updates to shared parameters can affect action distributions at states visited earlier in training.

We observe this effect in a synchronous GRPO WebShop run without KL regularization. We rescore the same 128 archived successful trajectories under five saved policies (Figure 2). From checkpoint 50 to checkpoint 149, all 128 trajectories lose likelihood, with a median paired decrease of 39.1% in length-normalized trajectory likelihood. This measurement tracks retention of recorded successful behavior; task success also depends on other trajectories that the current policy may generate. The fixed-cohort result reveals a within-run retention problem at the trajectory level: behavior that previously succeeded becomes less likely under later policies.

Policy-improvement theory provides a principled criterion for identifying historical trajectories as protection candidates. Let τ denote a complete interaction trajectory for task x, and let $R(\tau; x)$ be its verifiable, task-defined return. An oracle trajectory τ^* attains $R_{\max}(x) = \max_\tau R(\tau; x)$, so

$$A_\pi(\tau^* \mid x) := R_{\max}(x) - V_\pi(x) \geq 0, \qquad V_\pi(x) = \mathbb{E}_{\tau \sim \pi(\cdot|x)}[R(\tau; x)].$$

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.

This holds regardless of which policy generated τ^*. This justifies selecting historical oracle trajectories as protection targets. In practice, memory admission, sampling, and replay weights determine which trajectories contribute to the finite-buffer constraint and by how much.

We use A-GEM's aggregate half-space projection (Chaudhry et al., 2019) to protect historical successes when their gradient conflicts with the current objective. Let g be the main loss gradient and h the loss gradient of sampled successful replay. Success-Constrained Policy Gradient Projection (Figure 1) leaves g unchanged when $g^\top h \geq 0$ and otherwise makes the smallest Euclidean correction satisfying $h^\top v \geq 0$. The correction activates only when the main update would increase aggregate replay loss to first order. Successful replay has precedents in agentic RL (Lu et al., 2025; Xu et al., 2026). Our focus is retention within a single training run: we impose an aggregate constraint on historical successes and use A-GEM's projection geometry when the main gradient violates it.

Our contributions are threefold:

1. We identify and quantify within-run forgetting of previously successful trajectories in agentic RL under a fixed task distribution. A fixed cohort of 128 historical successes shows a 39.1% median decline in length-normalized trajectory likelihood in a WebShop training run.
2. We provide a policy-improvement rationale for selecting historical oracle trajectories as protection targets: their task-defined maximum return gives them nonnegative trajectory-level advantage relative to the current policy, regardless of which policy collected them. We define the empirical replay objective and apply A-GEM's aggregate projection when its gradient conflicts with the main gradient, quantifying the first-order cost.
3. Averaged over three training seeds, our method improves WebShop success by 6.2 percentage points with GRPO and 2.1 points with GiGPO, and raises GRPO's mean success above the GiGPO baseline without changing its advantage estimator. In a supplementary ALFWorld comparison using one training seed without policy collapse, it improves final success with both algorithms and reduces their decline from validation-selected saved checkpoints. Replay ablations compare FIFO and reservoir retention with uniform and prefix importance weights.

Section 2 relates our setting to policy optimization, continual learning, and success replay. Sections 3 and 4 derive the constraint and specify its protected distribution. Section 5 reports the experiments and retention diagnostics. The appendices give the oracle-event gradient identity, experimental protocol, and implementation details.

2 Setup and Related Work

2.1 Within-run retention setting

Throughout a training run, task prompts $x \sim p(x)$ are sampled from a fixed distribution, while the policy changes through optimization. Let π_θ denote the current policy and $\tau_i = (s_{i1}, a_{i1}, \ldots, s_{iJ_i}, a_{iJ_i})$ a trajectory collected earlier in the run under behavior policy μ_i. Here s_{ij} is the recorded context in which action a_{ij} was generated. A trajectory is oracle when its verifiable trajectory return reaches the task-defined maximum, including any action-format penalties; this criterion is distinct from the sign of a group-relative advantage. We assess retention by evaluating the likelihood of their recorded actions under π_θ. We seek to improve the current RL objective while preventing a first-order decrease in the aggregate weighted log likelihood of selected historical successes.

Let $L_{\text{base}}(\theta)$ denote the configured RL loss, including objective clipping and reference regularization where applicable. We use loss-gradient convention: $g = \nabla_\theta L_{\text{base}}(\theta)$ is the base gradient, and v is its corrected counterpart before the parameter update.

2.2 Related work

Continual learning and gradient constraints. GEM (Lopez-Paz & Ranzato, 2017) protects examples from each earlier task; A-GEM (Chaudhry et al., 2019) replaces task-wise constraints with an

aggregate memory gradient. CLEAR (Rolnick et al., 2019) mixes new and replayed experience in continual RL without requiring explicit task boundaries, but studies a sequence of tasks and retains broad past experience. CPO (Luo et al., 2026) likewise studies successive tasks and limits prior-task policy drift through a replay-free approximation to historical behavioral KL. In our setting, the task distribution and verifier remain fixed throughout training. We protect selected successful trajectories from earlier in the same run using A-GEM's projection geometry.

Replay of successful experience. Self-Imitation Learning (Oh et al., 2018) and advantage-weighted regression (Peng et al., 2019) reinforce high-return past actions. ARPO (Lu et al., 2025) inserts a stored success for the same task when a GRPO group contains only failures and has zero reward variance. MobileRL (Xu et al., 2026) addresses uneven task difficulty and rare successes with difficulty-adaptive positive replay, task filtering, and reward shaping. Self-imitating PPO (Chaudhary et al., 2026) also reuses successes; RRL (Dou et al., 2025) restarts exploration from promising prefixes that need not be completed successes. We study a different problem: previously successful behavior can lose likelihood while training continues on a fixed task distribution. We use historical successes as a retention target and correct an update when it conflicts with that target.

Off-policy correction. TRPO (Schulman et al., 2015), PPO (Schulman et al., 2017), and GRPO (Shao et al., 2024) use importance-sampling ratios $\pi_\theta/\pi_{\text{old}}$ in surrogate objectives to account for the difference between the rollout policy and the policy being optimized. These objectives use recently collected rollout data. A historical oracle trajectory may have been collected many updates earlier, yet its task-defined maximum return gives it nonnegative trajectory-level advantage relative to the current policy regardless of its behavior policy. This property justifies selecting off-policy oracle trajectories for the replay constraint.

3 Success-Constrained Policy Gradient Projection

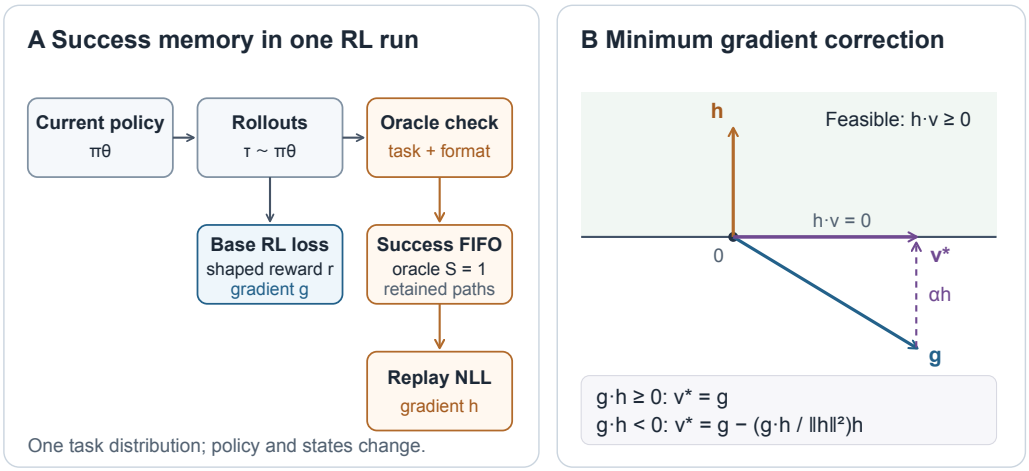


Figure 1: Success-Constrained Policy Gradient Projection. A task-defined oracle criterion selects trajectories for memory. At each update, the gradient h of their replay loss is compared with the base loss gradient g. If $g^\top h < 0$, the method finds the closest gradient v satisfying $h^\top v \geq 0$; otherwise $v = g$. The optimizer receives v.

3.1 Oracle trajectories and the replay objective

For a task prompt x, write $\tau = (s_1, a_1, \ldots, s_J, a_J)$ for a complete interaction trajectory, where s_j is the context before the agent generates action a_j. Let $p_\theta(\tau \mid x)$ be the trajectory distribution induced by the current policy and the environment. The task-defined return $R(\tau; x)$ includes any applicable penalties for invalid actions or response format. Define $R_{\max}(x) = \max_\tau R(\tau; x)$ and $S_x = \{\tau : R(\tau; x) = R_{\max}(x)\}$. For any $\tau^* \in S_x$, its trajectory-level advantage relative to the current policy is

$$A_\theta^{\mathrm{traj}}(\tau^*; x) = R_{\max}(x) - \mathbb{E}_{\tau \sim p_\theta(\cdot|x)}[R(\tau; x)] \geq 0.$$

This holds even when τ^* was generated by an earlier policy. If used as an action- or token-level training signal, A_θ^{traj} is shared across the trajectory and does not distinguish the contributions of individual actions or tokens.

We choose maximum-return trajectories, $\tau \in S_x$, as protection candidates. At one optimizer update, let $\mathcal{I}$ index the sampled trajectories from memory, with $K = |\mathcal{I}| > 0$. For each $i \in \mathcal{I}$, let T_i be the nonempty set of generated action-token positions (j, r) in trajectory i, where j indexes actions and r indexes tokens within an action. Let a_{ijr} be the token at that position and c_{ijr} its recorded context, including preceding generated tokens. Let $b_{ijr} \geq 0$ be a token-reduction coefficient and $w_{ij} \geq 0$ a detached action-level replay weight. With $w = \{w_{ij}\}$, define the replay loss $L_{\mathcal{I}}$ and its gradient h by

$$L_{\mathcal{I}}(\theta; w) = -\sum_{i \in \mathcal{I}} \sum_{(j,r) \in T_i} b_{ijr} w_{ij} \log \pi_\theta(a_{ijr} \mid c_{ijr}), \qquad h = \nabla_\theta L_{\mathcal{I}}(\theta; w).$$

For uniform replay, $w_{ij} = 1$ and $b_{ijr} = 1/(K|T_i|)$, so h averages the gradient of each trajectory's mean token negative log likelihood. For a fixed replay sample, the common factor $1/K$ only rescales h and leaves the zero-floor projection unchanged. The factors $1/|T_i|$ and any nonuniform w_{ij} set the relative contributions of trajectories and actions and can change the direction of h.

3.2 Minimum correction to the base gradient

Let $g = \nabla_\theta L_{\mathrm{base}}(\theta)$, following the loss-gradient convention of Section 2.1. To define the constraint, consider a gradient-descent step $\theta^+ = \theta - \gamma v$ with step size $\gamma > 0$ and candidate gradient v. Its first-order change in the replay loss is $-\gamma h^\top v$. We require $h^\top v \geq 0$ and choose the feasible gradient closest to g, using Euclidean distance over all trainable parameters:

$$v^* = \arg\min_v \frac{1}{2}\|v - g\|_2^2 \quad \text{subject to} \quad h^\top v \geq 0.$$

For $h \neq 0$, the solution is

$$v^* = g + \alpha h, \qquad \alpha = \left[-\frac{g^\top h}{\|h\|_2^2}\right]_+.$$

Here $[z]_+ = \max(z, 0)$. Thus $v^* = g$ when the gradients do not conflict, and $h^\top v^* = 0$ when they do. If $h = 0$ or memory contains no sample, we also set $v^* = g$. This is the aggregate half-space projection used by A-GEM (Chaudhry et al., 2019).

3.3 First-order effect and scope

Suppose g and h are nonzero and $g^\top h < 0$. Let $c = g^\top h/(\|g\|\|h\|)$ be the cosine of the angle between the two gradients. Projection gives

$$\frac{\|v^* - g\|}{\|g\|} = |c|, \qquad g^\top v^* = \|g\|^2(1 - c^2).$$

For the gradient-descent step above, the first-order decrease in the base loss changes from $\gamma\|g\|^2$ to $\gamma\|g\|^2(1 - c^2)$. Its relative first-order cost is therefore c^2 when projection activates and zero otherwise. The cost depends on the observed angle between g and h; it is not fixed by the method.

We impose a single constraint on the aggregate replay loss, so an individual stored trajectory may still lose likelihood. When $g^\top h < 0$, the replay-loss difference between the uncorrected and projected gradient-descent steps is

$$
\begin{aligned}
&L_{\mathcal{I}}(\theta - \gamma g; w) - L_{\mathcal{I}}(\theta - \gamma v^*; w) \\
&\quad = -\gamma h^\top (g - v^*) + o(\gamma) \\
&\quad = -\gamma h^\top g + o(\gamma).
\end{aligned}
$$

The second equality uses $h^\top v^* = 0$. Since $h^\top g < 0$, the leading term is positive: projection lowers replay loss relative to the uncorrected step to first order. At each update, the projection removes this component of the base gradient whenever the gradients conflict. This is the mechanism by which we aim to slow historical-success likelihood degradation.

3.4 Applying the correction

Once replay correction is active, at each optimizer update we sample oracle trajectories, compute their replay weights, and evaluate h at the current policy while holding those weights fixed. We compute g from the base RL loss, aggregate $g^\top h$ and $\|h\|^2$ across the trainable parameters, and replace g by v^* if the gradients conflict. The corrected gradient then enters the usual gradient-processing and optimizer steps. The base loss and its advantage estimator are unchanged.

Our primary implementation uses a memory warmup before activating the constraint. Oracle trajectories enter memory from the start, but we use the base gradient without projection until memory contains successes from a specified number of distinct prompts. This lets the first replay draws come from a broader set of successful prompts. Section 5 specifies the threshold used in the main experiments.

We project the loss gradient before optimizer processing, so the rule does not depend on a particular optimizer's update map. The condition $h^\top v^* \geq 0$ applies to a direct parameter step along $-v^*$; an optimizer may change that direction. Guaranteeing the same first-order property for the realized step would require constraining the optimizer's parameter displacement. Section 4 specifies which trajectories and weights enter $L_{\mathcal{I}}$; Appendix C details the implementation.

4 Defining the protected replay distribution

The projection in Section 3 is determined by the replay gradient h. Once oracle admission is fixed, two choices determine that gradient: **retention** decides which successful trajectories remain in memory, and **sampling** decides how retained actions contribute to the replay loss. We vary these choices independently.

4.1 Replay buffer retention

Each unique oracle trajectory is admitted as a whole to a buffer of capacity B. A FIFO buffer evicts the oldest trajectory when full, giving recent successes priority. This is a recency heuristic for reducing the gap between the behavior policies that collected the trajectories and the current policy, but it discards older successes.

A reservoir buffer instead gives each admitted success from the full stream the same retention probability. It represents the accepted stream in expectation and preserves broad temporal coverage. The retained trajectories can come from much older policies, whose context-arrival probabilities may differ from those of the current policy. The buffers contain the same trajectories before capacity is reached; their protected populations differ after eviction begins.

4.2 Sampling

At a replay update, we sample K whole trajectories uniformly without replacement from the buffer. For sampled trajectory i, let j index its actions, a_{ij} denote action j, s_{ij} its recorded pre-action context, and μ_i the policy that collected the trajectory. Uniform sampling sets every replay weight $w_{ij} = 1$. StateIS uses the same buffer draw but changes action-level contributions through the capped prefix ratio

$$r_{ij} = \prod_{k<j} \frac{\pi_\theta(a_{ik} \mid s_{ik})}{\mu_i(a_{ik} \mid s_{ik})}, \qquad w_{ij} = \min(C, r_{ij}), \quad C \geq 1.$$

Each action likelihood is the product of its generated-token likelihoods. The prefix product contains only actions $k < j$, so it is empty for $j = 1$ and $r_{i1} = 1$. Since $C \geq 1$, the capped weight is also $w_{i1} = \min(C, 1) = 1$.

All generated tokens in action j use the same weight w_{ij}. For a later action, the prefix includes more actions, so its weight may differ. StateIS changes the effective contribution of sampled actions, not the uniform draw of trajectories from memory.

The prefix ratio compares how likely the current and collection policies are to reach a recorded context. It reweights successes already admitted to memory; the cap C limits the influence of large ratios. Uniform weighting avoids this weight concentration, while the age of its sampled trajectories depends on the buffer's retention rule.

4.3 Coverage and weighting trade-offs

Retention and sampling jointly determine the temporal coverage of protected successes and the concentration of their replay weights. We compare the four configurations while holding oracle admission, capacity, token reduction, replay draw size, insertion timing, and zero-floor projection fixed. The table summarizes their expected properties:

Retention / Sampling	Uniform	StateIS
Reservoir	Broad historical coverage; older contexts receive unit weights	Broad coverage; prefix reweighting can concentrate weight on a few actions
FIFO	Recent successes; no ratio-induced weight concentration	Recent successes with prefix reweighting; its interaction with recency requires measurement

5 Experimental study

5.1 Protocol and primary comparison

We evaluate Success-Constrained Policy Gradient Projection with GRPO and GiGPO on WebShop (Yao et al., 2022) and ALFWorld (Shridhar et al., 2021). Our main WebShop study uses three independent training seeds. ALFWorld provides a supplementary comparison from one training seed in which none of the four compared runs exhibited policy collapse. Appendix B reports the seed values and analyzes ALFWorld policy collapse and late-training decline. For each benchmark, we compare each algorithm with its FIFO Uniform replay variant under the same training protocol. The replay arms keep the base loss and advantage estimator unchanged. Both benchmarks use Qwen2.5-1.5B-Instruct and a 150-batch training budget, with 16 prompt groups and eight rollouts per group (128 trajectories) in each batch. The designated final checkpoint is indexed 149. A collection batch can produce multiple optimizer updates because its trajectories are split into minibatches of action rows.

The primary replay configuration stores at most $B = 2048$ complete oracle trajectories and inserts successes from the current collection batch before learner updates. Oracle admission requires the task-defined maximum engineered return, including any invalid-format penalties. Each active replay update draws $K = 4$ trajectories uniformly without replacement and computes a fresh, equally trajectory-weighted replay gradient. We set the warmup threshold to 128 distinct successful prompts (Warm128). We use this replay configuration for both GRPO and GiGPO on both WebShop and ALFWorld.

For WebShop, we run the benchmark's 1K-product configuration with a fixed catalog realization, and assign goal IDs $[500, 6910)$ to training and $[0, 500)$ to final evaluation. GiGPO (Feng et al., 2025a) describes the full WebShop benchmark, while its released configuration (Feng et al., 2025b) defaults to the 1K-product version. Table 1 reports GRPO and GiGPO baselines rerun with our

fixed catalog and split. Our WebShop episodes allow up to 15 actions, with at most 512 generated tokens per action. Each checkpoint is evaluated with three decoding seeds at temperature 0.4, with one rollout per goal per decoding seed. Each evaluation requires exactly 500 distinct goals and the frozen catalog identity.

For ALFWorld, we train on its 3,553-game training split and use the full 140-game ValidSeen split for final evaluation. Episodes allow up to 50 actions and 512 generated tokens per action; prompts include the two most recent action-observation pairs. The task reward is 10 for completion and 0 otherwise, with a 0.1 penalty for invalid-format actions. Each checkpoint is evaluated with three decoding seeds at temperature 0.4, with one rollout per game per decoding seed. During training, each validation probe selects 128 distinct games from ValidSeen; final evaluation covers all 140 games.

5.2 WEBSHOP MAIN RESULTS

Table 1: Final-checkpoint First-500 success (%) under the matched WebShop protocol. Each training-seed cell is the mean ± sample standard deviation over three evaluation seeds, with 500 single-attempt goals per evaluation seed. The last column gives the mean ± sample standard deviation of the three training-checkpoint means.

Algorithm	Replay	Train seed 0	Train seed 1	Train seed 2	Across training seeds
GRPO	None	63.80 ± 1.11	67.73 ± 1.21	64.20 ± 0.72	65.24 ± 2.16
GRPO	FIFO Uniform, Warm128	70.40 ± 0.60	73.33 ± 0.42	70.47 ± 0.31	71.40 ± 1.67
GiGPO	None	70.27 ± 0.64	68.07 ± 1.36	72.40 ± 1.11	70.24 ± 2.17
GiGPO	FIFO Uniform, Warm128	73.00 ± 1.22	75.33 ± 0.61	68.80 ± 0.20	72.38 ± 3.31

All twelve training runs and all three evaluation seeds per checkpoint are complete. Replay improves mean success by 6.16 percentage points for GRPO and 2.14 points for GiGPO. GRPO improves in each training seed; GiGPO improves in seeds 0 and 1 and declines by 3.60 points in seed 2. GRPO with our correction reaches 71.40%, exceeding the GiGPO baseline at 70.24%. Thus protecting historical successes can yield gains comparable to finer-grained advantage estimation, while the same correction also improves GiGPO's mean performance.

5.3 ALFWORLD RESULTS AND LATE-TRAINING DECLINE

ALFWorld exhibits substantial variability across training runs, including policy collapse and pronounced late-training decline. Table 2 reports a supplementary comparison using one training seed for which none of the four compared runs exhibited policy collapse. Appendix B.4 analyzes policy collapse and failure trajectories at the 50-turn limit; Appendix B.5 details the late-training decline.

Table 2: Supplementary ALFWorld ValidSeen comparison using one training seed without policy collapse in any of the four runs. Success (%) is the mean ± sample standard deviation over three evaluation seeds. The best saved checkpoint is selected by validation success among checkpoints 25, 50, 75, 100, and 125, with ties resolved in favor of the later checkpoint. The final column gives the mean paired change from the best saved checkpoint to checkpoint 149 using matched evaluation seeds.

Algorithm	Replay	Best saved checkpoint	Best-checkpoint success	Final success	Final − best (pp)
GRPO	None	100	58.57 ± 5.00	32.62 ± 2.18	-25.95
GRPO	FIFO Uniform, Warm128	125	46.67 ± 2.97	36.67 ± 1.09	-10.00
GiGPO	None	125	65.95 ± 3.67	54.29 ± 0.71	-11.67
GiGPO	FIFO Uniform, Warm128	125	73.33 ± 2.06	70.00 ± 2.58	-3.33

Our method raises final success by 4.05 percentage points for GRPO and 15.71 points for GiGPO in this training seed. It also mitigates late-training decline: the decrease from the selected intermediate checkpoint is reduced from 25.95 to 10.00 points for GRPO and from 11.67 to 3.33 points for

GiGPO. All four policies still decline. Because checkpoint selection uses Seen validation and the selected steps differ across arms, these comparisons describe the observed training dynamics rather than comparing a common fixed interval. Across the replication campaign, two of twelve training runs were classified as collapsed; replay does not eliminate this instability.

5.4 HISTORICAL-SUCCESS LIKELIHOOD

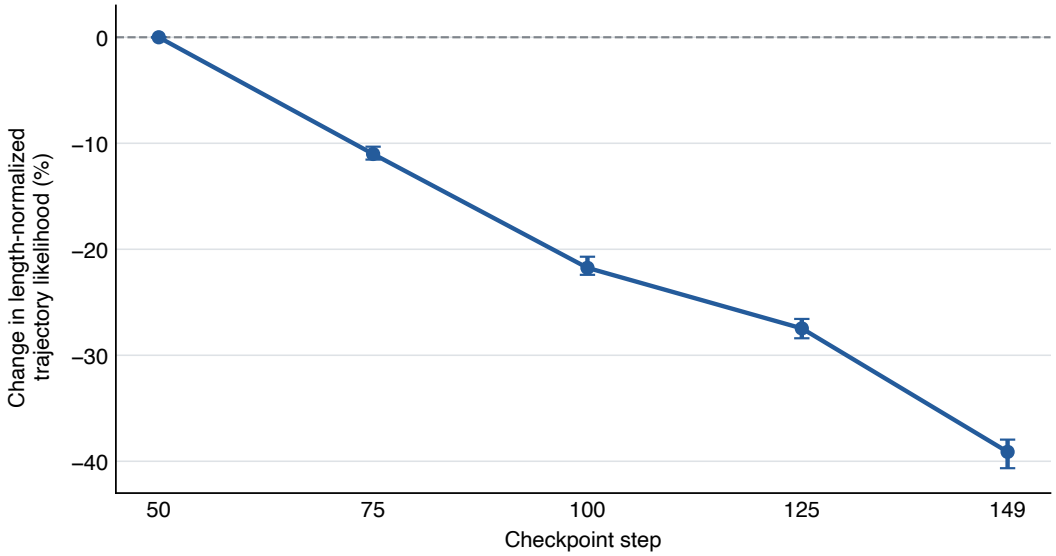


Figure 2: Fixed-cohort likelihood of historical successful trajectories. In a separate no-KL synchronous GRPO WebShop run, we score the same 128 successes, all collected before checkpoint 50, under five saved policies. For a trajectory with T generated tokens, define its length-normalized trajectory likelihood under checkpoint s as $\exp(L_s/T)$, where L_s is the sum of generated-token log probabilities. The vertical axis shows the median paired percentage change relative to checkpoint 50, $100[\exp((L_s - L_{50})/T) - 1]$. Bars show pointwise 95% trajectory-bootstrap intervals from 5,000 resamples, conditional on this training run.

The five checkpoints (50, 75, 100, 125, and 149) yield 640 scores for the fixed cohort, with the same trajectory identities and token counts throughout. Relative to checkpoint 50, median paired changes in length-normalized trajectory likelihood are -11.0%, -21.8%, -27.5%, and -39.1% at the four later checkpoints. At checkpoint 149, all 128 trajectories have lower length-normalized likelihood; the median has a 95% trajectory-bootstrap interval of $[-40.7\%, -38.0\%]$. These paired measurements show that the same successful trajectories become less likely over the course of training.

5.5 REPLAY ABLATIONS

To test the retention and sampling choices of Section 4, we use a WebShop GiGPO train-seed-0 factorial with FIFO or reservoir retention and Uniform or capped StateIS weighting ($C = 5$). All four cells share $B = 2048$, $K = 4$, Warm128, trajectory-uniform draws, oracle admission, and the same base training configuration.

Table 3: Final First-500 success (%) for the retention-by-sampling ablation at one training seed, mean $\pm$ sample standard deviation over three evaluation seeds. Difference intervals in the text describe evaluation randomness at fixed training checkpoints; Appendix B.6 specifies their calculation.

Retention	Uniform	StateIS
FIFO	73.00 ± 1.22	70.60 ± 0.53
Reservoir	76.40 ± 1.22	66.73 ± 0.90

Reservoir Uniform achieves the highest success, exceeding FIFO Uniform by 3.40 percentage points (pointwise 95% Welch interval: $[0.63, 6.17]$). Prefix weighting lowers success by 9.67 points under reservoir retention ($[-12.19, -7.15]$). Its 2.40-point decrease under FIFO has an interval crossing zero ($[-4.99, 0.19]$). These fixed-checkpoint comparisons favor reservoir retention with uniform weights and show a clear StateIS penalty under reservoir retention; the FIFO StateIS difference is less resolved by the three evaluation repeats. Trajectory-age and weight-concentration diagnostics would be needed to attribute these differences to particular distributional effects.

We retain FIFO Uniform as our default replay scheme: it directly prioritizes recent verified successes and requires no prefix-ratio weighting. Reservoir Uniform improves performance in this ablation, but its advantage has not been replicated across independent training seeds. The current comparison therefore supports evaluating broader historical coverage while retaining the intuitive recency-based scheme for the primary experiments.

6 Limitations and conclusion

Historical successful trajectories can lose likelihood during RL training even when the task distribution is fixed. Task-defined maximum-return trajectories have nonnegative trajectory-level advantage, providing a principled basis for selecting them as protection targets. Our projection removes first-order conflict between the main loss gradient and an aggregate loss on sampled successes. Across three training seeds on WebShop, our method improves mean performance with both GRPO and GiGPO, and raises GRPO above the GiGPO baseline without changing its advantage estimator. In the supplementary ALFWorld comparison using one training seed without policy collapse, it improves final success and mitigates the decline from selected intermediate checkpoints for both algorithms.

Our empirical study covers two environments and one model scale. ALFWorld's training variability and the single-seed replay ablation limit conclusions about robustness across runs. The theoretical constraint concerns aggregate first-order gradient geometry, while realized retention also depends on the replay distribution and optimizer dynamics. Broader evaluation and a better understanding of the interaction between success retention, exploration, and training stability are important directions for continual learning in agentic RL.

REPRODUCIBILITY STATEMENT

The experimental records include frozen evaluator settings, catalog identity, goal-level outcomes, runtime correction-mode checks, and configuration-conversion tests. Code and result artifacts have not yet been released anonymously.

AI USE STATEMENT

AI assistants were used in conceptual discussion, hypothesis and experiment design, implementation and debugging, log analysis and interpretation, literature discovery, mathematical exposition, and manuscript drafting and polishing.

REFERENCES

Gaurav Chaudhary, Laxmidhar Behera, and Washim Uddin Mondal. Match or Replay: Self Imitating Proximal Policy Optimization. *arXiv preprint arXiv:2603.27515*, 2026. URL https://arxiv.org/abs/2603.27515.

Arslan Chaudhry, Marc'Aurelio Ranzato, Marcus Rohrbach, and Mohamed Elhoseiny. Efficient Lifelong Learning with A-GEM. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=Hkf2_sC5FX.

Shihan Dou, Muling Wu, Jingwen Xu, Rui Zheng, Tao Gui, Qi Zhang, and Xuanjing Huang. Improving RL Exploration for LLM Reasoning through Retrospective Replay. *arXiv preprint arXiv:2504.14363*, 2025. URL https://arxiv.org/abs/2504.14363.

Lang Feng, Zhenghai Xue, Tingcong Liu, and Bo An. Group-in-Group Policy Optimization for LLM Agent Training. In *Advances in Neural Information Processing Systems*, volume 38, 2025a. URL https://papers.nips.cc/paper_files/paper/2025/hash/420c9f777c0b4f78d515e53cf74d58b2-Abstract-Conference.html.

Lang Feng, Zhenghai Xue, Tingcong Liu, and Bo An. verl-agent: WebShop training configuration, 2025b. URL https://github.com/langfengQ/verl-agent/blob/f974dc5977908d6007ad178c9f3c48f0c7b39331/verl/trainer/config/ppo_trainer.yaml. Software configuration, commit f974dc5977908d6007ad178c9f3c48f0c7b39331.

David Lopez-Paz and Marc'Aurelio Ranzato. Gradient Episodic Memory for Continual Learning. In *Advances in Neural Information Processing Systems*, volume 30, 2017. URL https://papers.nips.cc/paper/2017/hash/f87522788a2be2d171666752f97ddebb-Abstract.html.

Fanbin Lu, Zhisheng Zhong, Shu Liu, Chi-Wing Fu, and Jiaya Jia. ARPO: End-to-End Policy Optimization for GUI Agents with Experience Replay. *arXiv preprint arXiv:2505.16282*, 2025. URL https://arxiv.org/abs/2505.16282.

Mao-Lin Luo, Zhe-Xu Wang, Zi-Hao Zhou, Bo Ye, Jian Zhao, Min-Ling Zhang, and Tong Wei. RL Forgets! Towards Continual Policy Optimization. *arXiv preprint arXiv:2607.04364*, 2026. URL https://arxiv.org/abs/2607.04364.

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. PMLR, 2018. URL https://proceedings.mlr.press/v80/oh18b.html.

Xue Bin Peng, Aviral Kumar, Grace Zhang, and Sergey Levine. Advantage-Weighted Regression: Simple and Scalable Off-Policy Reinforcement Learning. *arXiv preprint arXiv:1910.00177*, 2019. URL https://arxiv.org/abs/1910.00177.

David Rolnick, Arun Ahuja, Jonathan Schwarz, Timothy Lillicrap, and Gregory Wayne. Experience Replay for Continual Learning. In *Advances in Neural Information Processing Systems*, volume 32, 2019. URL https://papers.nips.cc/paper_files/paper/2019/hash/fa7cdfad1a5aaf8370ebeda47a1ff1c3-Abstract.html.

John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust Region Policy Optimization. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pp. 1889–1897. PMLR, 2015. URL https://proceedings.mlr.press/v37/schulman15.html.

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.

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://openreview.net/forum?id=OIOX0YcCdTn.

Avi Singh, John D. Co-Reyes, Rishabh Agarwal, Ankesh Anand, Piyush Patil, Xavier Garcia, Peter J. Liu, James Harrison, Jaehoon Lee, Kelvin Xu, Aaron Parisi, Abhishek Kumar, Alex Alemi, Alex Rizkowsky, Azade Nova, Ben Adlam, Bernd Bohnet, Gamaleldin Elsayed, Hanie Sedghi, Igor Mordatch, Isabelle Simpson, Izzeddin Gur, Jasper Snoek, Jeffrey Pennington, Jiri Hron, Kathleen Kenealy, Kevin Swersky, Kshiteej Mahajan, Laura Culp, Lechao Xiao, Maxwell L. Bileschi, Noah Constant, Roman Novak, Rosanne Liu, Tris Warkentin, Yundi Qian, Yamini Bansal, Ethan Dyer, Behnam Neyshabur, Jascha Sohl-Dickstein, and Noah Fiedel. Beyond Human Data: Scaling Self-Training for Problem-Solving with Language Models. *Transactions on Machine Learning Research*, 2024. URL https://openreview.net/forum?id=lNAyUngGFK.

Yifan Xu, Xiao Liu, Xinghan Liu, Jiaqi Fu, Hanchen Zhang, Bohao Jing, Shudan Zhang, Yuting Wang, Wenyi Zhao, and Yuxiao Dong. MobileRL: Online Agentic Reinforcement Learning for Mobile GUI Agents. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=C3F0G9nXhl.

Shunyu Yao, Howard Chen, John Yang, and Karthik Narasimhan. WebShop: Towards Scalable Real-World Web Interaction with Grounded Language Agents. In *Advances in Neural Information Processing Systems*, volume 35, 2022. URL https://papers.nips.cc/paper_files/paper/2022/hash/82ad13ec01f9fe44c01cb91814fd7b8c-Abstract-Conference.html.

A Oracle-event gradient and projection geometry

A.1 Current-policy oracle event

For a fixed task prompt x, let S_x be the set of maximum-return trajectories from Section 3.1 and define $J_x(\theta) = \Pr_{\tau\sim p_\theta(\cdot|x)}(\tau \in S_x) > 0$. The task verifier and environment dynamics do not depend on θ. Differentiating this event probability gives

$$\begin{aligned}\nabla_\theta \log J_x(\theta) &= \frac{1}{J_x(\theta)} \sum_{\tau\in S_x} p_\theta(\tau \mid x)\nabla_\theta \log p_\theta(\tau \mid x) \\ &= \mathbb{E}_{\tau\sim q_\theta(\cdot|x,S_x)}[\nabla_\theta \log p_\theta(\tau \mid x)],\end{aligned}$$

where $q_\theta(\tau \mid x, S_x) = p_\theta(\tau \mid x)\mathbf{1}_{S_x}(\tau)/J_x(\theta)$. The same identity holds with an integral for a continuous trajectory space. It describes the direction obtained by sampling oracle trajectories from the *current* policy conditional on the oracle event. Our replay gradient instead uses historical oracle trajectories retained and sampled according to Section 4. The nonnegative trajectory-level advantage justifies their selection; the buffer and weights determine the empirical gradient.

A.2 Alignment with an oracle-event target

Let $t = -\nabla_\theta J_x(\theta)$ be the loss-gradient target for increasing the oracle-event probability. For a nonzero replay gradient h, write $n = h/\|h\|$ and $r = [-g^\top n]_+$. The projected gradient is $v = g + rn$. Direct expansion gives

$$\|v - t\|^2 - \|g - t\|^2 = -r^2 - 2r\, n^\top t.$$

Thus, when $n^\top t \geq 0$, projection does not increase squared error relative to t, and strictly reduces it if the constraint activates. The exact current-policy oracle-event loss gradient $h_* = -\nabla_\theta \log J_x(\theta)$ satisfies $t = J_x(\theta)h_*$, so it has this alignment. For a finite historical replay gradient, alignment is an empirical question; the projection rule itself requires only g and h.

B Experimental protocol and supplementary measurements

B.1 Matched primary arms

The primary comparison crosses GRPO and GiGPO with no replay or FIFO Uniform replay. Both methods use group-wise advantage normalization in the matched runs. GRPO uses episode-outcome advantages; GiGPO combines step-level and episode-level advantages with weights 1:1 and step discount 0.95. Replay changes neither estimator. The replay arms use a capacity of 2,048 whole trajectories, draw four trajectories per active optimizer update, and wait until 128 distinct successful prompts are stored before activating projection.

Table B1: Matched training and evaluation settings for the primary arms.

Protocol element	WebShop	ALFWorld
Training prompts	Goal IDs 500–6909 in a fixed 1K-product catalog (seed 233)	3,553 training games
Collection budget	150 batches; 16 prompts × 8 rollouts per batch	150 batches; 16 prompts × 8 rollouts per batch
Episode limit	15 actions; at most 512 generated tokens per action	50 actions; at most 512 generated tokens per action
Training seeds with completed primary runs	0, 1, 2	0, 1, 2; two collapsed runs were not evaluated
Final evaluation	First-500 goals, one rollout per goal for each decoding seed 123, 456, 789	ValidSeen 140 games, one rollout per game for each decoding seed 0, 1, 2

The designated final checkpoint is batch 149, and final rollouts use temperature 0.4. All twelve WebShop primary runs have complete three-seed final evaluations. ALFWorld has complete three-seed final evaluations for ten of twelve primary runs, including all four seed-1 arms; Appendix B.4 identifies the two exclusions. WebShop validation probes use goals outside First-500, whereas ALFWorld validation probes draw 128 games from the 140-game ValidSeen split used for final evaluation.

B.2 WebShop replay controls

The following controls use the matched WebShop train-seed-1 protocol. Each value is final First-500 success, mean ± sample standard deviation over three decoding seeds.

Table B2: WebShop replay controls at training seed 1.

Replay rule	GRPO	GiGPO
FIFO Uniform, $K = 4$, no warmup	$70.3 \pm 0.2\%$	$64.2 \pm 0.8\%$
FIFO Uniform, $K = 4$, Warm128	$73.3 \pm 0.4\%$	$75.3 \pm 0.6\%$

The two rows differ in whether the Warm128 activation threshold is applied. These controls are separate from the WebShop GiGPO train-seed-0 retention-by-sampling factorial in Section 5.5.

B.3 Fixed-cohort likelihood scoring

The Figure 2 cohort contains 128 archived successful WebShop trajectories collected at batches 2–49 of one no-KL synchronous GRPO run. Each trajectory is scored under checkpoints 50, 75, 100, 125, and 149, yielding 640 scores with fixed trajectory identities and generated-token counts. For trajectory i, its paired percentage change at checkpoint s is

$$d_{is} = 100 \left[\exp\left(\frac{L_{is} - L_{i,50}}{N_i} \right) - 1 \right],$$

where L_{is} is the sum of current-policy log probabilities of its recorded generated tokens and N_i is its generated-token count. Figure 2 plots the median of d_{is} across the 128 trajectories. Each pointwise interval resamples trajectory identities with replacement 5,000 times and recomputes that median. The interval summarizes variation across this fixed cohort within one training run.

B.4 ALFWorld policy collapse and horizon failures

We examined training dynamics for the four primary arms at training seeds 0, 1, and 2. All twelve runs reached batch 149, but GRPO baseline seed 2 and GiGPO replay seed 0 exhibited policy collapse during training. These two policies were not evaluated at the final checkpoint. Scheduler completion therefore does not imply successful policy learning, and collapse occurs with and without replay. The supplementary comparison in Table 2 uses training seed 1, for which none of the four runs exhibited policy collapse. The following trajectory analysis examines how ineffective behavior produces horizon failures; Section B.5 examines late-training decline in the four noncollapsed runs.

Failure trajectories at the 50-turn limit. A separate, earlier exact-once diagnostic of the seed-1 GRPO baseline at checkpoint 149 evaluated 140 Seen games with decoding seed 0. It recorded 41 successes and 99 failures; all 99 failures reached the 50-turn cap. This diagnostic is separate from the three-seed final aggregate in Table 2. Reaching the cap describes an episode failure mode; it is not by itself a criterion for declaring an entire training run collapsed.

The worker conversations expose actions 1–49 for all 99 failed episodes and 40 of the 41 successful episodes; terminal action 50 is not present in a subsequent prompt. Among failed episodes, 84/99 repeat the same nonempty navigation command for their first ten actions, 97/99 contain an empty executed action, and 96/99 contain at least five empty actions. Of the 4,851 observed failed-episode actions, 2,335 are nonempty but absent from the recorded admissible-action list, and 1,431

are empty. Together these account for 77.6% of observed actions. The median first-empty-action position is 28 among failures containing an empty action. Only 25/99 failed episodes visibly pick up the task target within the observed actions. These traces show repeated ineffective actions and late response-format deterioration among horizon failures.

B.5 ALFWorld late-training decline

Checkpoint selection and paired decline. Before the paired evaluation, we selected the highest-validation checkpoint among saved checkpoints 25, 50, 75, 100, and 125 for each seed-1 arm, resolving ties in favor of the later checkpoint. Selection used the original training-validation records; an unsaved validation peak was not substituted. We then evaluated that checkpoint and checkpoint 149 with the same three decoding seeds on Seen140. For evaluation seed e, let S_e^{int} and S_e^{149} be the two success rates. The paired change is $\Delta_e = S_e^{149} - S_e^{\text{int}}$, in percentage points.

Table B3: ALFWorld seed-1 paired checkpoint changes. The last column reports the mean $\pm$ sample standard deviation of Δ_e over the three evaluation seeds.

Algorithm	Replay	Best saved checkpoint	Best-checkpoint success (%)	Final success (%)	Paired change (pp)
GRPO	None	100	58.57 ± 5.00	32.62 ± 2.18	-25.95 ± 4.76
GRPO	FIFO Uniform, Warm128	125	46.67 ± 2.97	36.67 ± 1.09	-10.00 ± 3.98
GiGPO	None	125	65.95 ± 3.67	54.29 ± 0.71	-11.67 ± 4.19
GiGPO	FIFO Uniform, Warm128	125	73.33 ± 2.06	70.00 ± 2.58	-3.33 ± 3.67

All four seed-1 arms lose task success after their selected checkpoint, with smaller declines under replay. Since selection uses Seen validation, this is a comparison to each run's selected intermediate performance, not a fixed-step comparison or an estimate of an unbiased peak. These task-level changes complement the trajectory-likelihood diagnostic in Section 5.4, which evaluates a fixed set of historical successes.

B.6 Uncertainty in replay ablations

All intervals in Section 5.5 are pointwise, two-sided 95% intervals over evaluation randomness on the fixed First-500 goals and fixed trained checkpoints. Each ablation uses one training seed; these intervals do not measure variability across training runs or across task distributions. They are not simultaneous intervals for the full set of comparisons.

For Warm128, we pair success rates by decoding seed 123, 456, and 789. Let d_e be Warm128 success minus no-warmup success, in percentage points. The interval is $\bar{d} \pm t_{0.975,2} s_d/\sqrt{3}$, where s_d is the sample standard deviation of the three differences. GRPO's paired differences are 3.6, 2.6, and 2.8 points; GiGPO's are 11.0, 11.8, and 10.6 points.

For the retention-by-sampling contrasts, the available audit summaries provide complete three-seed means and sample standard deviations but not all paired seed vectors. We therefore use Welch intervals without estimating cross-arm covariance. For arms A and B, the standard error is $\sqrt{s_A^2/3 + s_B^2/3}$, with Welch–Satterthwaite degrees of freedom. Input summaries are rounded to two decimal places, so the reported intervals are approximate. Both calculations use Student-t assumptions with only three repeats.

Table B4: Replay ablation effects and pointwise 95% confidence intervals (percentage points). A positive effect favors the first condition named in the contrast.

Contrast	Mean effect	95% interval	Calculation
GRPO: Warm128 − no warmup	3.00	$[1.69, 4.31]$	Paired Student-t
GiGPO: Warm128 − no warmup	11.13	$[9.62, 12.65]$	Paired Student-t
Uniform: Reservoir − FIFO	3.40	$[0.63, 6.17]$	Welch
FIFO: StateIS − Uniform	−2.40	$[-4.99, 0.19]$	Welch
Reservoir: StateIS − Uniform	−9.67	$[-12.19, -7.15]$	Welch

C REPLAY IMPLEMENTATION

C.1 CONSTRUCTING THE REPLAY GRADIENT

The task-defined oracle test is applied to complete trajectories, including any penalties for invalid actions or format. Admitted trajectories enter the buffer before learner updates on their collection batch. FIFO evicts the oldest whole trajectory when capacity is reached; the retention factorial also tests reservoir insertion. At an active replay update, the implementation draws K whole trajectories uniformly without replacement and scores their recorded agent-generated action tokens under the current policy. Environment-response tokens are not replay loss targets.

For Uniform sampling, every action weight is one and each generated token in trajectory i has reduction coefficient $1/(K|T_i|)$, using the notation of Section 3.1. The replay gradient is recomputed for each optimizer update. In the StateIS factorial, tokens within action j share its detached, capped prefix weight from Section 4.2, with cap $C = 5$. Scaling all weights in a fixed draw by the same positive constant does not change the zero-floor projection; changing their relative values can change the replay-gradient direction.

The projection compares the base and replay loss gradients before subsequent gradient processing and the optimizer step. It is one aggregate constraint on the sampled replay loss, rather than a separate constraint for every stored trajectory.