BALANCING LEARNABILITY AND BELLMAN-ERROR COVERAGE IN PRIORITIZED EXPERIENCE REPLAY

Anonymous authors
Paper under double-blind review

ABSTRACT

Experience replay improves the sample efficiency of off-policy reinforcement learning, but its benefit depends on which transitions are revisited. Prioritized experience replay (PER) favours large temporal difference (TD) errors, which may overemphasize noisy or irreducible samples. Reducible-loss prioritization (ReLo) instead targets errors that appear learnable, yet it can fail in the complementary direction: when the reference loss is imperfect, difficult but informative transitions may receive negligible priority and disappear from replay. We introduce BLCR (balanced learnability–coverage replay), a conservative repair that combines a normalized TD-error priority floor with bounded reducible-loss weighting. Unlike an additive mixture, the floor is activated only when ReLo underprioritizes a high-error transition; otherwise, the reducible-loss ranking is preserved. Across five MinAtar games at two million frames and five seeds, BLCR outperforms ReLo on every game and obtains the highest final return on Asterix and Breakout. Across 30 ALE-RAM games, it outperforms ReLo on 25 games and wins 118 of 150 paired-seed comparisons. In a preliminary 300K-step SAC extension over four MuJoCo locomotion tasks, BLCR-SAC ranks first on Hopper and Walker2d, improving the final return over the SAC by 12.8% and 56.5%, respectively, while the results on Ant and HalfCheetah remain mixed. These findings position conservative TD-error coverage as a targeted correction to reducible-loss replay rather than a universal replacement for PER.

1 INTRODUCTION

Off-policy deep reinforcement learning derives much of its data efficiency from experience replay: transitions collected under earlier policies can be reused for multiple value-function updates, reducing temporal correlation and amortizing the cost of environmental interaction (Lin, 1992; Mnih et al., 2015). Replay, however, introduces a decision that is as consequential as the update rule itself: Given a finite training budget, which stored transitions deserve to be revisited? Uniform replay treats all the samples equally, even though their learning values can vary substantially over the course of training.

Prioritized experience replay (PER) addresses this question by sampling transitions in proportion to their TD error magnitude (Schaul et al., 2016). The heuristic is effective because a large TD error often indicates a transition on which the current value function requires correction. The same signal is also ambiguous. A transition may retain a large TD error because of stochastic rewards, partial observability, nonstationary bootstrap targets, or approximation error that the learner cannot readily reduce. Repeatedly replaying such samples can result in updates on persistent error rather than useful learning progress.

Reducible-loss prioritization (ReLo) offers a more selective principle: prioritize a sample according to the portion of its loss that appears reducible by the learner (Sujit et al., 2023). In our value-based instantiation, this signal compares the online network loss with a reference loss produced by the target network. A transition receives high priority when the online network performs worse than the reference does and low priority when both incur similar errors. This distinction is attractive because it separates *being wrong* from *being learnably wrong*, filtering samples whose large losses may be irreducible.

The selectivity of ReLo can nevertheless result in the opposite failure mode to that of PER. The reference model is only a proxy for irreducibility: it is delayed, changes throughout training, and

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.

shares the same function class and bootstrapped targets as the learner does. Consequently, a small estimated reducible loss does not imply that a transition has become unimportant. When this estimate collapses for transitions that still carry large Bellman errors, their replay probabilities can become negligible. The resulting *TD error undercoverage* is a plausible explanation for the large ReLo deficits observed in several MinAtar and ALE-RAM tasks; establishing this mechanism causally requires the replay diagnostics discussed later in the paper.

A natural response is to add TD error back into the ReLo score. This is unsatisfactory both conceptually and empirically. An unconstrained additive mixture allows TD error to alter every ranking and to dominate whenever its scale increases, gradually turning the method back into PER and reintroducing its sensitivity to noisy outliers. The desired correction is asymmetric: TD error should protect a transition from vanishing, but it should not amplify a transition that ReLo already considers sufficiently learnable.

Our method, BLCR (balanced learnability–coverage replay), implements this asymmetric correction. Let $\widehat{R}_i$ denote the normalized nonnegative reducible loss of transition i, and let $\widehat{D}_i$ denote its normalized absolute TD error. BLCR assigns priority

$$p_i = \max\Big\{\widehat{R}_i,\ \lambda \log(1+\widehat{D}_i)\Big\}, \tag{1}$$

where λ controls a compressed TD error floor. The maximum operator is the key design choice: if the reducible loss already provides sufficient priority, the TD error has no effect; if the reducible-loss signal becomes too small while the Bellman error remains large, the floor restores replay coverage. Within the sampled mini-batch, BLCR further applies a bounded weight

$$g_i = \mathrm{clip}\Big(1+\eta \log(1+\widehat{R}_i),\ 1,\ g_{\max}\Big) \tag{2}$$

to the per-sample Bellman loss. This second mechanism modestly strengthens the updates on samples identified as learnable while preventing unbounded gradient amplification. Together, the two components separate *coverage* from *update strength*: the floor determines which difficult transitions remain visible, whereas the bounded weight determines how strongly learnable sampled transitions affect optimization.

The empirical evidence supports a focused claim. In five MinAtar games (Young & Tian, 2019), when two million frames and five seeds are used, BLCR improves the last-100-episode return of ReLo on all five tasks and is the strongest evaluated method on Asterix and Breakout. On a broader 30-game ALE-RAM benchmark derived from the Arcade Learning Environment (ALE) (Bellemare et al., 2013), BLCR exceeds ReLo on 25 games and records a higher return in 118 of 150 paired-seed comparisons. These gains are not obtained by universally imitating PER: PER remains stronger on some MinAtar tasks, most notably Seaquest, while the floor primarily repairs cases in which ReLo appears to suppress useful correction samples.

We also examine whether the same replay principle transfers beyond discrete-action value learning. For this preliminary soft actor-critic (SAC) study (Haarnoja et al., 2018), learnability is measured by the realized one-step reduction in per-sample critic loss; the TD-error floor and bounded critic-loss weight are otherwise unchanged. This local proxy is shared by the ReLo-SAC and BLCR-SAC variants in our comparison but differs from the target-critic proxy used in the original ReLo SAC experiments. Across four MuJoCo locomotion tasks at 300K steps (Todorov et al., 2012), BLCR-SAC achieves the highest final evaluation return on Hopper and Walker2d. It reaches 3105.6 ± 135.7 on Hopper versus 2754.0 ± 727.8 for SAC and 3201.5 ± 841.9 on Walker2d versus 2046.0 ± 395.9 for SAC. The remaining tasks define a clear boundary: BLCR-SAC is comparable to SAC but below ReLo-SAC on Ant, and it trails SAC on HalfCheetah at 300K steps.

This work makes the following contributions:

- We formulate TD-error undercoverage as a failure mode of reducible-loss replay: an imperfect learnability proxy can suppress transitions that remain important for Bellman correction.
- We derive BLCR as the minimum change to the ReLo priorities required to satisfy a compressed TD-error coverage constraint and pair this floor with bounded reducible-loss weighting.
- We establish the scope of the method through five-seed MinAtar and 30-game ALE-RAM evaluations while reporting failure cases and controlled floor-only ablation.

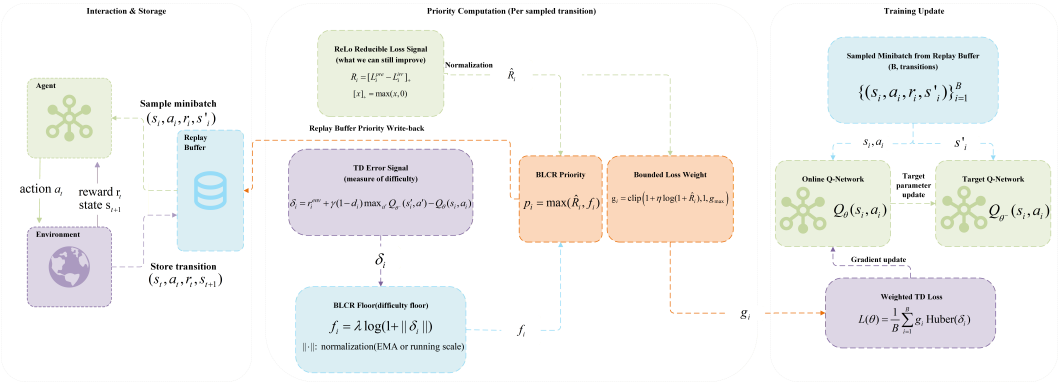

Figure 1: BLCR replay updating pipeline. The interaction stage stores the transitions in replay; the priority-computation stage combines the normalized ReLo signal $\widehat{R}_i$ with the TD-error floor f_i and writes the resulting priority back to the buffer; and the training stage uses the sampled mini-batch to update the online network with a bounded weighted TD loss. The diagram uses compact notation for readability; the full equations below include normalization, priority clipping, importance-sampling correction, and bounded factor definitions.

- We provide an extension to the SAC and evidence of transfer to selected locomotion tasks, without claiming universal superiority across environments or training horizons.

Figure 1 summarizes the resulting replay-update pipeline. Interaction first fills the replay buffer; each sampled transition is then rescored by combining the normalized ReLo reducible-loss signal with a compressed TD-error floor; finally, the updated priority controls future replay, while a bounded weight scales the present Bellman update.

2 Related Work

2.1 Experience Replay and Replay Distributions

Experience replay was introduced to reuse past interactions and decorrelate online updates (Lin, 1992); its combination with a target network became central to the DQN (Mnih et al., 2015). Subsequent work has shown that replay performance depends not only on the buffer capacity but also on the replay ratio, age distribution, and interaction between replay and the underlying learning algorithm (Fedus et al., 2020). These observations motivate treating the replay distribution as an optimization decision rather than a passive data-management detail.

PER assigns a transition a priority proportional to its absolute TD error and corrects the resulting sampling bias with annealed importance weights (Schaul et al., 2016). Its effectiveness has made prioritized replay a component of integrated agents such as Rainbow (Hessel et al., 2018). TD error is inexpensive and directly tied to the Bellman objective, but it does not distinguish learnable error from persistent noise. Large batch experience replay (LaBER) instead draws a large candidate batch and downsamples according to approximate gradient norms, interpreting replay as variance reduction for stochastic optimization (Lahire et al., 2022). LaBER improves the freshness of priorities but requires scoring more candidates than are ultimately used. BLCR addresses a different issue: it constrains a selective learnability-based priority so that high-error transitions cannot vanish while scoring only the training mini-batch.

2.2 Reliability, Learnability, and Corrective Weighting

Several methods modify the training distribution to reduce harmful bootstrapping or distributional mismatch. DisCor reweights Bellman updates using an estimated error model so that target errors do not compound unchecked (Kumar et al., 2020). ReLo instead estimates the learnable portion of each sample's loss. In deep Q-learning, it uses a lagged target network as a reference model and

defines priority from the positive difference between online and reference losses (Sujit et al., 2023). Samples on which both models have high loss are interpreted as difficult or noisy and are therefore downprioritized.

BLCR is built on ReLo but changes its failure semantics. ReLo asks whether the online learner is worse than its reference; BLCR additionally asks whether the underlying Bellman residual is still large. The TD signal is not added globally. It is imposed as a one-sided floor, so it changes the ReLo priority only when a coverage constraint is violated. This distinguishes BLCR from convex or additive combinations of multiple priority scores and from methods that directly replace priority with gradient magnitude.

2.3 Replay in Actor–Critic Learning

Off-policy actor–critic methods also depend on replay, but their nonstationary actor and twin critics make the choice of learnability proxy consequential. SAC combines a stochastic actor, entropy regularization, twin critics, and slowly updated target critics (Haarnoja et al., 2018). Prioritized replay can be applied to critic TD errors, while a ReLo-style score can be formed from either a reference critic or an observed loss reduction. The original ReLo SAC implementation uses the target critic as its reference; our pilot instead uses realized one-step loss reduction for both ReLo-SAC and BLCR-SAC. The actor, target-network, and temperature updates are identical across our SAC variants; thus, the comparison isolates replay sampling and critic loss weighting under this common proxy.

3 Method

3.1 Preliminaries

Consider a discounted Markov decision process with state space $\mathcal{S}$, action space $\mathcal{A}$, transition kernel P, reward r, and discount γ. A replay buffer $\mathcal{D}$ contains transitions $x_i = (s_i, a_i, r_i, s'_i, d_i)$, where $d_i \in \{0, 1\}$ indicates termination. For value-based learning, an online action-value network Q_θ and target network $Q_{\bar{\theta}}$ define

$$y_i = r_i + \gamma(1 - d_i) \max_{a'} Q_{\bar{\theta}}(s'_i, a'), \tag{3}$$

$$\delta_i = y_i - Q_\theta(s_i, a_i), \qquad D_i = |\delta_i|. \tag{4}$$

We use the elementwise Huber loss $h(\delta)$ and write the online loss as $L_i^{\text{on}} = h(\delta_i)$. Uniform replay samples every stored transition with probability $1/N$, where $N = |\mathcal{D}|$. For a nonnegative replay score p_i, including $p_i = D_i$ for PER, our implementation samples according to

$$P(i) = \frac{(p_i + \epsilon_p)^\alpha}{\sum_{j=1}^{N} (p_j + \epsilon_p)^\alpha}, \tag{5}$$

where $\alpha \in [0, 1]$ controls the prioritization strength and $\epsilon_p > 0$ preserves support for every stored transition.

Following ReLo, the target network also evaluates the current state–action pair under the same target:

$$L_i^{\text{ref}} = h(y_i - Q_{\bar{\theta}}(s_i, a_i)), \qquad R_i = \left[L_i^{\text{on}} - L_i^{\text{ref}}\right]_+. \tag{6}$$

R_i is large when the online network has loss that the lagged reference does not share. It is small both when the sample is already fitted and when the online and reference models fail similarly. In the second case, a learnability estimate can erase a transition that still has a substantial Bellman residual.

3.2 Scale-Normalized Signals

The magnitudes of R_i and D_i differ across games and training stages. Combining unnormalized signals would make λ environmentally dependent. We therefore maintain a separate exponential moving-average (EMA) scale for each nonnegative component $X \in \{R, D\}$. For mini-batch $\mathcal{B}_t$, let $\mathcal{B}_{t,X}^+ = \{i \in \mathcal{B}_t : X_i > 0,\ X_i \text{ finite}\}$. When this set is nonempty,

$$m_{X,t} = \rho m_{X,t-1} + (1 - \rho) \frac{1}{|\mathcal{B}_{t,X}^+|} \sum_{i \in \mathcal{B}_{t,X}^+} X_i, \tag{7}$$

with the first positive batch mean used for initialization. The normalized component is

$$\widehat{X}_i = \mathrm{clip}\left(\frac{X_i}{\max(m_{X,t}, \epsilon_n)}, 0, c_{\max}\right). \tag{8}$$

Here, $\rho \in [0, 1)$ is the EMA decay, $\epsilon_n > 0$ prevents division by zero, and $c_{\max} > 0$ caps outliers. Separate scales prevent one signal from overwhelming the other; clipping preserves ordering below $c_{\max}$.

3.3 Balanced Learnability–Coverage Priority

We seek the smallest modification of the ReLo priority that guarantees a TD-dependent minimum. Define the compressed coverage requirement

$$f_i = \lambda \log(1 + \widehat{D}_i), \qquad \lambda \geq 0. \tag{9}$$

For each transition, consider the constrained projection

$$\min_{q_i \geq 0} \frac{1}{2}(q_i - \widehat{R}_i)^2 \quad \text{subject to} \quad q_i \geq f_i. \tag{10}$$

Its unique solution is $q_i^\star = \max\{\widehat{R}_i, f_i\}$. Thus, the BLCR priority is

$$p_i^{\mathrm{BLCR}} = \mathrm{clip}\Big(\max\Big\{\widehat{R}_i, \lambda \log(1 + \widehat{D}_i)\Big\}, p_{\min}, p_{\max}\Big). \tag{11}$$

where $0 < p_{\min} < p_{\max}$ is the bound numerical priority. Equation equation 10 makes the unclipped maximum unique feasible priority closest to $\widehat{R}_i$. Equation equation 11 preserves that projection whenever $q_i^\star \leq p_{\max}$, as in our configuration, while $p_{\min}$ enforces support. Unlike an additive score, BLCR otherwise changes only floor-active transitions satisfying $f_i > \widehat{R}_i$.

Two immediate properties clarify the behavior. First, in the unclipped regime, if $\widehat{R}_i \geq f_i$, then $p_i^{\mathrm{BLCR}} = \widehat{R}_i$; therefore, ReLo priorities and pairwise rankings are preserved among floor-inactive transitions. Second, define the clipped floor $\widetilde{f}_i = \mathrm{clip}(f_i, p_{\min}, p_{\max})$. The monotonicity of the clipping gives $p_i^{\mathrm{BLCR}} \geq \widetilde{f}_i$. Since every score is at most $p_{\max}$, Eq. equation 5 gives the per-update lower bound

$$P(i) \geq \frac{(\widetilde{f}_i + \epsilon_p)^\alpha}{N(p_{\max} + \epsilon_p)^\alpha}. \tag{12}$$

The bound does not guarantee fast learning, but it prevents a high-error transition from receiving an arbitrarily small probability solely because the reducible-loss estimate is near zero.

3.4 Bounded Learnability Weighting

Sampling controls which transitions enter a mini-batch, whereas the objective controls their effect after selection. We use a second EMA-normalized copy $\widehat{R}_i^\ell$ and define

$$g_i = \mathrm{clip}\Big(1 + \eta \log(1 + \widehat{R}_i^\ell), 1, g_{\max}\Big), \tag{13}$$

where $\eta \geq 0$ controls the weighting strength and $g_{\max} \geq 1$ is its upper bound. The normalized PER importance weight is

$$\omega_i = \frac{(NP(i))^{-\beta}}{\max_{j \in \mathcal{B}}(NP(j))^{-\beta}}, \tag{14}$$

with β annealed from β_0 to 1. The DQN objective is

$$\mathcal{L}_{\mathrm{BLCR}}(\theta) = \frac{1}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} \omega_i g_i h(\delta_i). \tag{15}$$

The importance weight corrects the nonuniform sampling distribution, whereas g_i is an intentional change to the optimization objective. Because $0 < \omega_i \leq 1$ and $1 \leq g_i \leq g_{\max}$, the total per-sample multiplier is bounded by $g_{\max}$. This prevents the learnability term from creating unbounded gradient amplification. In all the discrete experiments, $\rho = 0.99$, $\lambda = 0.35$, $\eta = 0.08$, $g_{\max} = 1.25$, $c_{\max} = p_{\max} = 10$, $p_{\min} = \epsilon_p = 10^{-5}$, and $\epsilon_n = 10^{-6}$.

Algorithm 1 One discrete-action BLCR update

Require: Replay buffer $\mathcal{D}$, online network Q_θ, target network $Q_{\bar{\theta}}$

1: Sample indices $i \sim P$ and compute normalized IS weights ω_i
2: Compute targets y_i, TD magnitudes D_i, and online losses L_i^{on}
3: Compute reference losses L_i^{ref} and $R_i = \left[L_i^{\text{on}} - L_i^{\text{ref}}\right]_+$
4: Normalize R_i for the loss and compute bounded factors g_i using Eq. equation 13
5: Update θ by minimizing Eq. equation 15
6: Normalize R_i and D_i with their priority EMAs
7: Set $p_i \leftarrow \text{clip}(\max\{\widehat{R}_i, \lambda \log(1 + \widehat{D}_i)\}, p_{\min}, p_{\max})$
8: Update replay priorities; if synchronization is due, set $\bar{\theta} \leftarrow \theta$

3.5 Extension to Soft Actor–Critic

For the SAC, we retain the actor, entropy-temperature, and target-network updates and modify only the critic replay. The learnability signal is the realized one-step decrease in twin-critic loss, $R_i^{\text{SAC}} = [L_i^- - L_i^+]_+$, while the coverage signal is the postupdate mean absolute critic TD error D_i^+. After mini-batch normalization, the future replay priority is $\max\{\widehat{R}_i^{\text{SAC}}, \lambda \log(1 + \widehat{D}_i^+)\} + \epsilon_p$; the currently available preupdate TD magnitude supplies the bounded weight for the present critic step. Appendix A gives the complete targets, normalization, and weighting equations. This realized-reduction proxy is shared by ReLo-SAC and BLCR-SAC in our pilot and is not presented as a reproduction of the original target-critic ReLo estimator.

4 Experiments and Analysis

Table 1: MinAtar last-100-episode return after two million frames (mean ± standard deviation over five seeds). **Bold** indicates the highest mean in each game. ReLo-F is the priority-floor ablation without bounded loss weighting.

Game	DQN	PER	ReLo	ReLo-F	BLCR
Asterix	14.52 ± 0.83	15.21 ± 2.24	14.07 ± 1.82	17.14 ± 0.54	$\mathbf{17.89 \pm 1.84}$
Breakout	8.64 ± 0.96	9.79 ± 1.20	8.43 ± 0.43	9.62 ± 1.03	$\mathbf{10.59 \pm 0.34}$
Freeway	51.64 ± 1.33	$\mathbf{51.91 \pm 0.52}$	51.13 ± 0.77	50.89 ± 1.53	51.54 ± 0.52
Seaquest	17.71 ± 4.19	$\mathbf{28.42 \pm 4.35}$	18.22 ± 3.35	19.33 ± 2.18	23.11 ± 6.85
Space Invaders	37.25 ± 1.59	$\mathbf{42.15 \pm 4.00}$	28.72 ± 1.99	39.72 ± 4.58	40.43 ± 1.72

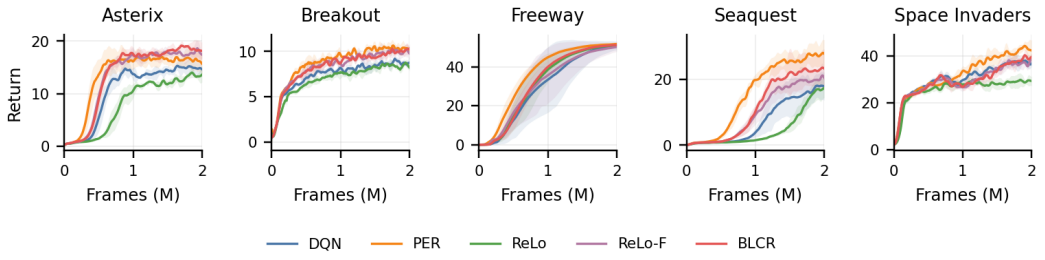

Figure 2: MinAtar learning curves over two million frames. Lines show the across-seed mean of the logged average training return, and shaded regions show one standard deviation across five seeds. BLCR consistently repairs ReLo deficits, whereas PER remains strongest on some tasks.

4.1 Evaluation Questions and Protocol

The experiments address four questions: (Q1) does the TD floor recover performance lost by ReLo; (Q2) does bounded loss weighting add value beyond the floor; (Q3) does the effect persist over a broad task set; and (Q4) does the same principle transfer to continuous-control critics? Every completed

task in each stated suite is included under a fixed aggregation rule. Following established concerns about variance in deep RL evaluation (Henderson et al., 2018; Agarwal et al., 2021), the tables report the mean $\pm$ standard deviation across seeds wherever space permits, and the final-performance statistics are kept distinct from the best-checkpoint scores.

For MinAtar, each method is trained for two million environment frames with seeds 0–4. The shared DQN uses one 3×3 convolution with 16 channels, a 128-unit hidden layer, Huber loss, RMSProp with a learning rate 2.5×10^{-4}, a batch size of 32, a 100K replay buffer, and a target update every 1K frames. Exploration decreases from 1.0 to 0.1 over 100K frames. For prioritized methods, $\alpha = 0.6$ and β are annealed from 0.4 to 1. The primary metric is the per-seed mean return over the last 100 training episodes, followed by the mean and standard deviation across seeds.

The ALE experiment uses RAM observations on 30 games, two million frames, and five seeds per method. Actions are repeated for four emulator frames, sticky actions are disabled, training rewards are clipped to $[-1, 1]$, and a two-layer 256-unit MLP is used. This suite is a broad paired comparison between ReLo and BLCR; it does not contain uniform-DQN or PER baselines and is interpreted accordingly. The continuous control method uses a SAC with two 256-unit hidden layers, a batch size of 256, a one-million-transition buffer, $\gamma = 0.99$, target smoothing $\tau = 0.005$, automatic entropy tuning, and learning rates of 3×10^{-4}. Training begins after 5K steps and performs one update per environment step. Prioritized SAC uses $\alpha = 0.6$ and anneals β from 0.4 to 1; BLCR-SAC uses $\lambda = 0.35$, $\eta = 0.08$, $g_{\max}^{\mathrm{SAC}} = 1.5$, $c_{\max} = 10$, and $\epsilon_p = 10^{-6}$. Appendix B reports the classic-control boundary cases, and Figure 5 in the appendix illustrates the benchmark families.

4.2 MinAtar: Repairing ReLo Without Replacing PER

Relative to ReLo, BLCR improves all five MinAtar means by 27.1% on Asterix, 25.7% on Breakout, 0.8% on Freeway, 26.9% on Seaquest, and 40.8% on Space Invaders (Table 1 and Figure 2). The floor-only ablation recovers most of the gap in Asterix, Breakout, and Space Invaders, and BLCR is numerically greater than ReLo-F on every game. The positive logged priority lift in Figure 4(b) confirms that the floor changes the sampled priorities in practice, although the current logs do not identify the activation rate or replay mass of the rescued transitions.

The same results rule out uniform dominance. PER remains strongest on Freeway, Seaquest, and Space Invaders: Freeway is nearly saturated, whereas Seaquest and Space Invaders are consistent with either insufficient floor strength at $\lambda = 0.35$ or a setting where the raw TD error is the most effective replay signal. Thus, BLCR improves this ReLo implementation without eliminating task dependence.

4.3 ALE-RAM: Breadth across 30 Games

Across Table 3, BLCR is higher for 25 of 30 games, ReLo is higher for four, and Montezuma's Revenge is tied at zero. At the seed level, BLCR had a higher paired return in 118 of 150 comparisons. The greatest relative gains occur in the Battle Zone ($2.42\times$), Alien ($2.13\times$), Hero ($2.00\times$), Demon Attack ($1.86\times$), and Amidar ($1.85\times$). The method does not merely improve games with small reward scales: absolute gains are also large on Battle Zone, Crazy Climber, Hero, and Krull. Representative positive, negative, and saturated cases are shown in Figure 3.

Bank Heist and Kung Fu Master favour ReLo, Fishing Derby degrades slightly, and Freeway remains effectively tied. Montezuma's Revenge shows that replay coverage cannot create reward information that the behavior policy never discovers. The broad win count supports Q3—the improvement is not confined to the five MinAtar games—, but the absence of DQN and PER in this 30-game suite prevents a claim of overall state of the art. The experiments establish improvement over the ReLo parent method, which is not superior to all replay strategies.

4.4 Continuous-Control Transfer

At 300K steps, BLCR-SAC ranks first on Hopper and Walker2d, improving over the SAC by 12.8% and 56.5%, respectively. It also has much lower across-seed dispersion than does the SAC on Hopper. On Ant, it is essentially tied with SAC and below ReLo-SAC; on HalfCheetah, it is 8.6% below SAC.

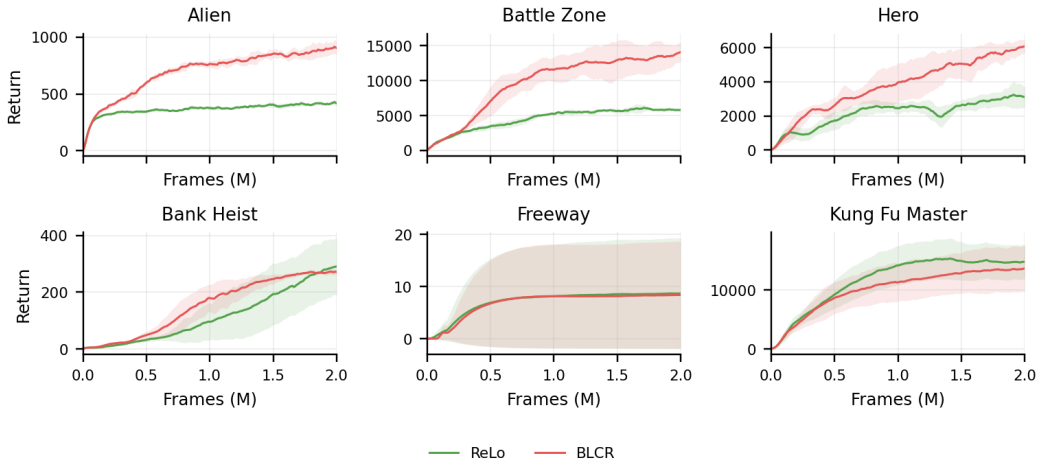

Figure 3: Representative ALE-RAM learning curves. Alien, Battle Zone, and Hero show large BLCR gains; Bank Heist and Kung Fu Master show cases where ReLo is competitive or stronger; Freeway illustrates a near-saturated task with little room for replay-distribution differences.

Table 2: Deterministic MuJoCo evaluation returns at 300K steps (mean ± standard deviation over three seeds). Each evaluation averages five deterministic episodes per seed. ReLo-SAC denotes our realized one-step loss-reduction proxy, not a reproduction of the original target-critic ReLo estimator. The best mean for each task is in bold.

Method	Ant	Half Cheetah	Hopper	Walker2d
SAC	1336.2 ± 254.4	$\mathbf{7499.0 \pm 488.9}$	2754.0 ± 727.8	2046.0 ± 395.9
PER-SAC	1329.7 ± 387.2	6486.5 ± 879.2	2211.3 ± 697.7	1128.1 ± 750.5
ReLo-SAC	$\mathbf{1462.3 \pm 499.9}$	7040.1 ± 589.9	1881.3 ± 926.0	881.1 ± 541.8
BLCR-SAC	1336.7 ± 150.0	6854.4 ± 507.5	$\mathbf{3105.6 \pm 135.7}$	$\mathbf{3201.5 \pm 841.9}$

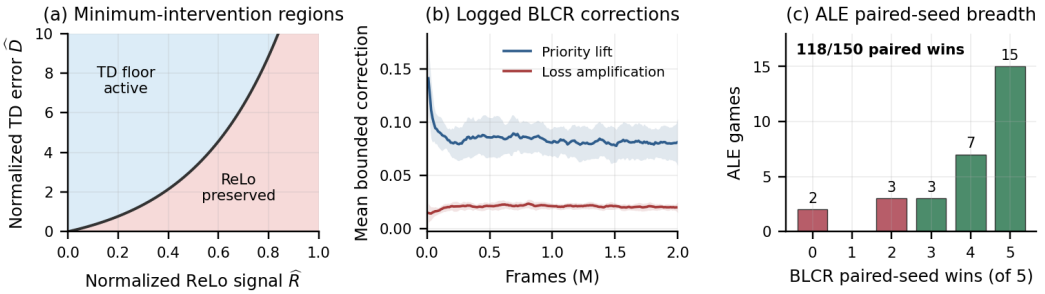

Figure 4: Mechanism geometry and nonreturn diagnostics. (a) The TD floor changes a ReLo priority only below the boundary $\widehat{R} = \lambda \log(1 + \widehat{D})$. (b) Batch-averaged priority lift $\max(p_i - \widehat{R}_i, 0)$ and bounded loss amplification $g_i - 1$, averaged over all 25 MinAtar game-seed runs; shading denotes one standard deviation across runs. Both corrections remain nonzero and bounded during training. (c) Distribution of task-level paired-seed wins on ALE-RAM: BLCR wins all five seeds on 15 games and 118 of 150 pairs overall. Panels (b)–(c) use logged training diagnostics and final last-100 returns, respectively, rather than additional reward curves.

These mixed results answer Q4 conditionally: the mechanism transfers to some continuous-control critics but does not provide an environment-independent advantage.

4.5 When Does the Correction Help?

Three patterns emerge. First, the gains are greatest when ReLo is weak: Space Invaders in MinAtar and Alien, Battle Zone, Hero, and Ms. Pac-Man in ALE-RAM yield large recoveries. This is consistent with a repair mechanism because the floor is inactive whenever ReLo already exceeds its constraint. In Figure 4(a), this one-sided intervention is explicit, whereas in panel (b), the resulting priority lift remains positive but bounded throughout MinAtar training. Second, near-saturated tasks such as Freeway offer little room for any replay rule, whereas exploration-limited Montezuma's Revenge cannot benefit from replaying undiscovered rewards. Third, continuous-control gains concentrate on Hopper and Walker2d at 300 K, whereas Ant and HalfCheetah mark current boundary cases in which the fixed floor and loss factor do not improve the SAC critic.

These correlations do not establish causal TD-error undercoverage: the diagnostics measure average intervention, not which transitions are rescued. Testing the mechanism requires per-update floor activation $\Pr[f_i > \widehat{R}_i]$, rescued sampling mass, and subsequent loss reduction. Rescued mass followed by learning progress would support the mechanism; activation without progress would instead suggest noisy-error replay. These measurements would distinguish the proposed explanation from generic regularization.

5 Conclusions and Limitations

BLCR repairs reducible-loss replay with a minimum-intervention TD-error floor and bounded loss weighting, separating coverage from update strength. It improves over ReLo on all five MinAtar games and 25 of 30 ALE-RAM games (118 of 150 seed pairs), with SAC gains concentrated on Hopper and Walker2d at 300K steps. These results support a targeted correction, not uniform superiority over PER or uniform replay.

These findings do not establish state-of-the-art ALE performance: the 30-game study compares only ReLo and BLCR, uses RAM, and disables sticky actions. MinAtar covers five games, with negative cases in classic control. Continuous-control evidence is limited to three seeds and 300K steps, without real-robot validation. The fixed λ and η lack exhaustive sensitivity analysis; undercoverage remains a hypothesis rather than a directly verified mechanism. Finally, the prioritized SAC prototype uses an avoidable $O(N)$ sampler, limiting throughput despite mini-batch-linear signal computation.

Future work should measure floor activation and postreplay progress, compare pixel-based ALE against uniform replay and PER, extend continuous-control budgets, and report stratified bootstrap intervals. Deployment claims require safety-constrained simulator-to-hardware tests with rare disturbances, comparing return, constraint violations, recovery success, and replayed-event coverage under an identical SAC controller.

AI Use Statement

Generative AI tools were used only for experimental code implementation and manuscript language polishing. The main methodological ideas and the analysis of the experimental results were entirely the authors' own work. The authors independently reviewed the AI-assisted code and text and take full responsibility for all claims, analyses, and content in this submission.

Reproducibility Statement

The method is specified by the equations and pseudocode in Algorithm 1, while the experimental protocol, aggregation rules, and hyperparameters are described in the evaluation section. An anonymized code repository containing the implementation and experimental launch configurations needed to reproduce the reported comparisons is available at https://anonymous.4open.science/r/BLCR-FA8B. The complete task-level results and additional details are provided in the appendix.

REFERENCES

Rishabh Agarwal, Max Schwarzer, Pablo Samuel Castro, Aaron C. Courville, and Marc G. Bellemare. Deep reinforcement learning at the edge of the statistical precipice. In *Advances in Neural Information Processing Systems*, volume 34, pp. 29304–29320, 2021.

Marc G. Bellemare, Yavar Naddaf, Joel Veness, and Michael Bowling. The arcade learning environment: An evaluation platform for general agents. *Journal of Artificial Intelligence Research*, 47: 253–279, 2013.

William Fedus, Prajit Ramachandran, Rishabh Agarwal, Yoshua Bengio, Hugo Larochelle, Mark Rowland, and Will Dabney. Revisiting fundamentals of experience replay. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 3061–3071, 2020.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *Proceedings of the 35th International Conference on Machine Learning*, pp. 1861–1870, 2018.

Peter Henderson, Riashat Islam, Philip Bachman, Joelle Pineau, Doina Precup, and David Meger. Deep reinforcement learning that matters. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, pp. 3207–3214, 2018.

Matteo Hessel, Joseph Modayil, Hado van Hasselt, Tom Schaul, Georg Ostrovski, Will Dabney, Dan Horgan, Bilal Piot, Mohammad Gheshlaghi Azar, and David Silver. Rainbow: Combining improvements in deep reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, pp. 3215–3222, 2018.

Aviral Kumar, Abhishek Gupta, and Sergey Levine. DisCor: Corrective feedback in reinforcement learning via distribution correction. In *Advances in Neural Information Processing Systems*, volume 33, pp. 18560–18572, 2020.

Thibault Lahire, Matthieu Geist, and Emmanuel Rachelson. Large batch experience replay. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 11790–11813, 2022.

Long-Ji Lin. Self-improving reactive agents based on reinforcement learning, planning and teaching. *Machine Learning*, 8:293–321, 1992.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin Riedmiller, Andreas K. Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015.

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

Shivakanth Sujit, Somjit Nath, Pedro H. M. Braga, and Samira Ebrahimi Kahou. Prioritizing samples in reinforcement learning with reducible loss. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5026–5033, 2012.

Kenny Young and Tian Tian. Minatar: An atari-inspired testbed for thorough and reproducible reinforcement learning experiments, 2019.

A Additional Method Details

A.1 BLCR-SAC Equations

For the SAC, let Q_{θ_1} and Q_{θ_2} be twin critics, π_ϕ be the actor, and α_{ent} be the learned entropy temperature. With $a'_i \sim \pi_\phi(\cdot|s'_i)$, the target is

$$y_i = r_i + \gamma(1-d_i)\Big[\min_{k\in\{1,2\}} Q_{\bar{\theta}_k}(s'_i, a'_i) - \alpha_{\text{ent}} \log \pi_\phi(a'_i|s'_i)\Big]. \tag{16}$$

Let $L_i^- = \frac{1}{2}\sum_{k=1}^{2}(Q_{\theta_k}(s_i,a_i) - y_i)^2$ be the twin-critic loss before an update and L_i^+ its value afterward; the same $-/+$ convention applies to the critic outputs. We define

$$R_i^{\text{SAC}} = \left[L_i^- - L_i^+\right]_+, \tag{17}$$

$$D_i^+ = \frac{1}{2}\sum_{k=1}^{2} |Q_{\theta_k}^+(s_i,a_i) - y_i|. \tag{18}$$

Each nonnegative SAC signal X_i is normalized within the current mini-batch:

$$\widehat{X}_i = \text{clip}\left(\frac{X_i}{\max\left(|\mathcal{B}|^{-1}\sum_{j\in\mathcal{B}} X_j, \epsilon_n\right)}, 0, c_{\max}\right). \tag{19}$$

The future replay priority is

$$p_i^{\text{BLCR-SAC}} = \max\left\{\widehat{R}_i^{\text{SAC}}, \lambda \log(1+\widehat{D}_i^+)\right\} + \epsilon_p. \tag{20}$$

Because realized loss reduction is available only after the critic step, the current update uses $D_i^- = \frac{1}{2}\sum_{k=1}^{2}|Q_{\theta_k}(s_i,a_i) - y_i|$ to compute

$$g_i^{\text{SAC}} = \text{clip}\left(1 + \eta \log(1+\widehat{D}_i^-), 1, g_{\max}^{\text{SAC}}\right). \tag{21}$$

The realized reduction determines which samples are replayed next, whereas the currently available TD signal weights the present critic update. These computations are $O(|\mathcal{B}|)$ and require one postupdate evaluation of both critics.

B Additional Experimental Results

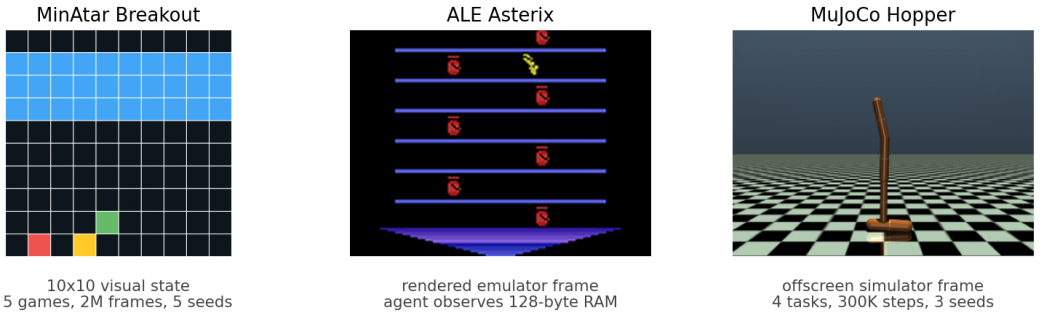

Figure 5: Representative rendered frames from the evaluation domains. MinAtar uses a compact Breakout grid, ALE-RAM observes 128-byte RAM from the Asterix emulator, and MuJoCo uses continuous-control Hopper states.

B.1 Complete ALE-RAM Results

Table 3: Mean last-100-episode return across five seeds on all 30 ALE-RAM games. Only ReLo and BLCR were run in this suite. Higher values are better; the higher mean in each pair is in bold.

Game	ReLo	BLCR	Game	ReLo	BLCR
Alien	427.64	**911.16**	Demon Attack	157.15	**292.09**
Amidar	60.55	**112.16**	Enduro	28.37	**40.66**
Assault	351.68	**379.19**	Fishing Derby	**-88.84**	-91.00
Asterix	917.90	**1261.80**	Freeway	**8.70**	8.39
Asteroids	1150.88	**1568.32**	Frostbite	170.90	**189.80**
Atlantis	30860.40	**31917.20**	Gopher	951.88	**1279.80**
Bank Heist	**293.76**	270.72	Gravitar	245.10	**379.30**
Battle Zone	5862.00	**14172.00**	Hero	3071.44	**6140.03**
Beam Rider	503.12	**602.41**	Ice Hockey	-7.85	**-6.87**
Bowling	32.06	**41.65**	James Bond	60.90	**80.20**
Boxing	55.38	**75.92**	Kangaroo	614.20	**620.40**
Breakout	11.17	**15.29**	Krull	5494.56	**6901.56**
Centipede	2359.28	**3219.48**	Kung Fu Master	**14598.80**	13625.20
Chopper Command	918.40	**1348.00**	Montezuma's Revenge	0.00	0.00
Crazy Climber	37101.40	**41821.00**	Ms. Pac-Man	699.72	**1210.04**

B.2 Classic Control Boundary Cases

Table 4: Gymnasium classic-control last-100 return after one million steps (mean across three seeds). Higher is better.

Task	DQN	PER	ReLo	BLCR
Acrobot	-87.33	-86.05	-94.20	$\mathbf{-85.88}$
CartPole	101.51	127.94	$\mathbf{182.34}$	109.01
LunarLander	115.72	11.83	$\mathbf{155.02}$	107.31
MountainCar	-200.00	-200.00	-200.00	-200.00

Table 4 deliberately broadens the evidence beyond favourable domains. BLCR is best on Acrobot, but ReLo is substantially stronger on CartPole and LunarLander, and no method solves MountainCar under this configuration. These low-dimensional tasks have small replay buffers and rapidly changing policies; a fixed floor may retain stable high-error transitions after they cease to be useful. This is consistent with BLCR as a repair for undercoverage rather than a default replacement whenever ReLo already provides adequate coverage.

B.3 Computational Cost

Across the completed 300K MuJoCo runs, the mean throughput is 83.5 frames/s for SAC, 69.5 for PER-SAC, 66.5 for ReLo-SAC, and 68.5 for BLCR-SAC. Most of the gap is shared by all prioritized variants and comes from the prototype's array-wide probability normalization and categorical sampling over a growing replay buffer. BLCR-SAC additionally performs the postupdate critic evaluation needed to measure realized loss reduction. Replacing the continuous replay backend with a segment tree or batched alias sampler removes the $O(N)$ sampling bottleneck without changing the optimization equations or hyperparameters. Until this engineering change is validated for numerical equivalence, the wall-clock efficiency remains a limitation of the current implementation.