Proceedings of the 4th International Conference on Machine Learning and Automation
DOI: 10.54254/2755-2721/2026.38508

Regulating Replay Concentration in Prioritized Experience Replay Through Normalized Replay Effective Support

Yize Zheng

Academy of Artificial Intelligence and Advanced Technology, Xi'an Jiaotong-Liverpool University, Suzhou, China
Yize.Zheng23@student.xjtlu.edu.cn

Abstract. This study investigates whether replay concentration in prioritized experience replay can be measured and regulated directly through normalized replay effective support, and how such regulation affects reinforcement-learning performance across tasks. Fixed-exponent prioritized replay is first examined in three low-dimensional control environments and five MinAtar games to characterize how realized replay support varies across tasks and training stages. An adaptive controller is then introduced to adjust the priority exponent so that replay support satisfies a prescribed floor. Replay concentration is evaluated using effective support together with entropy, probability-mass concentration, and sampling-diversity diagnostics, while learning performance is assessed using final return and time-normalized area under the evaluation curve. Experiments show that a fixed priority exponent does not correspond to a consistent replay-support state across tasks, whereas adaptive control can maintain a common support target with high recalibration accuracy and limited computational overhead. However, learning outcomes remain strongly task dependent. Clear improvements are observed in some MinAtar environments, particularly Seaquest, while other tasks show limited, inconclusive, or unfavorable outcomes relative to uniform replay and established prioritization baselines. These findings show that replay effective support is useful for characterizing and regulating sampling concentration, but accurate control of the replay distribution alone does not guarantee improved learning performance.

Keywords: prioritized experience replay, reinforcement learning, replay concentration, effective support, adaptive control

1. Introduction

Experience replay lets off-policy reinforcement-learning agents learn from transitions collected earlier. With uniform replay, every transition in the buffer has the same chance of being sampled, regardless of how much it may contribute to learning. Prioritized Experience Replay (PER) uses temporal-difference (TD) errors to assign sampling priorities [1]. The priority exponent α determines how strongly sampling favors transitions with higher priorities.

Even when α stays fixed, the distribution of replay probabilities can change. If priorities are close in value, sampling remains spread across the buffer. If a few priorities are much larger than the rest,

© 2026 The Authors. This is an open access article distributed under the terms of the Creative Commons Attribution License 4.0 (https://creativecommons.org/licenses/by/4.0/).

DOI: 10.54254/2755-2721/2026.38508

those transitions may receive much of the sampling probability. TD errors evolve during training and differ across environments, so the exponent alone does not describe how concentrated replay has become. Measuring the resulting distribution provides this additional information.

Earlier work has examined both the coverage of the replay buffer and the learning signals used to assign priorities. Schaul et al. noted that strong prioritization can reduce replay diversity [1], and Pan et al. examined outdated priorities and insufficient sample-space coverage [2]. Lahire et al. studied the theoretical basis of replay sampling and analyzed PER as an approximation to an importance-sampling objective [3]. High TD error is not necessarily a reducible learning signal [4]; other approaches adjust priorities dynamically [5] or account for unreliable target estimates [6]. Replay-sampling entropy has also been investigated [7]. These findings give reasons to examine the replay distribution, but the effects of explicitly controlling its concentration across tasks remain an empirical question.

This paper used Normalized Replay Effective Support (NRES) to measure this concentration. NRES is the inverse of the sum of squared replay probabilities, divided by buffer size. Before this normalization, the measure is the order-two Hill number [8]. Related effective-sample-size criteria are also used in adaptive sequential Monte Carlo [9]. Here, NRES serves as a diagnostic of PER and as a target for controlling replay support. This paper examined whether the target can be reached, how support drifts between recalibrations, and the computational cost of control, alongside learning outcomes in different tasks. NRES describes the effective support of the replay proposal distribution. It does not estimate how many training samples are statistically independent.

This study investigates whether replay concentration in Prioritized Experience Replay can be measured and regulated directly through replay effective support, and how such regulation affects learning performance across tasks. First, fixed-exponent PER is examined across three low-dimensional control environments and five MinAtar games to determine how replay support varies across tasks and training stages and how NRES relates to complementary concentration and sampling-diversity measures. Second, an adaptive controller adjusts the priority exponent to maintain a prescribed support floor, with target feasibility, inter-control drift, and computational overhead evaluated explicitly. Third, controlled replay is compared with uniform replay and established prioritization baselines to investigate its effects on learning performance. The analysis distinguishes successful regulation of the replay distribution from improvements in return.

2. Method

2.1. Environments and experimental setup

Experiments use Double DQN [10] on CartPole-v1, Acrobot-v1, and LunarLander-v3 for 50,000, 50,000, and 120,000 environment steps, respectively. Across these tasks, $\gamma = 0.99$, one gradient update is performed per step after warm-up, the target network is synchronized every 1,000 steps, and β increases linearly from 0.4 to 1.0. Online and target networks use two 128-unit ReLU hidden layers. Adam uses a learning rate of 5×10^{-4}, replay capacity 10,000, minibatches of 64, and a 1,000-step warm-up. Exploration decreases from 1.0 to 0.01 over 40,000 steps. Updates use importance-weighted squared TD error with gradient clipping at 10. Fixed PER with $\alpha \in \{0.6,1.0\}$ characterizes replay concentration.

Final low-dimensional performance comparisons include uniform replay, task-tuned PER, ReaPER [6], and NRES-controlled PER using ten matched test seeds. Fixed PER is tuned independently on five development seeds that do not overlap with the test set. The candidate exponents are $\{0.2,0.4,0.6,0.8,1.0\}$, and selection is based on mean evaluation-curve AUC. This

DOI: 10.54254/2755-2721/2026.38508

procedure selects $\alpha = 0.6$ for CartPole, $\alpha = 1.0$ for Acrobot, and $\alpha = 0.2$ for LunarLander. ReaPER follows the authors' public replay-buffer update semantics under the same Double-DQN backbone, using $\alpha = 0.4$ and reliability exponent $\omega = 0.2$. Exploratory controller experiments use three seeds per setting. Control intervals $K \in \{500,1000,2500\}$ are evaluated at $\tau = 0.20$, while a separate target-response sweep considers $\tau \in \{0.15,0.20,0.25,0.30,0.40\}$ with $K = 500$. The final controller setting, $\tau = 0.30$ and $K = 500$, is selected from tracking, feasibility, and computational-overhead diagnostics rather than from return maximization.

MinAtar experiments [11] use Asterix, Breakout, Freeway, Seaquest, and Space Invaders with the full six-action set. Because these tasks provide spatial observations and use a one-million-frame training horizon, benchmark-specific architecture and replay settings are adopted rather than transferring the low-dimensional configuration unchanged. Networks use sixteen convolutional filters followed by a 128-unit fully connected layer. Centered RMSProp with a learning rate of 2.5×10^{-4} is used as a fixed DQN-style visual-control optimization setting rather than being tuned separately for NRES. Replay capacity is 100,000, minibatch size is 32, and warm-up lasts 5,000 frames. MinAtar updates use importance-weighted Huber loss without gradient clipping. Exploration decreases from 1.0 to 0.1 over 100,000 frames, while the discount factor, target-update period, update frequency, and β schedule remain unchanged. All MinAtar settings are fixed before the formal comparison and shared by uniform replay, fixed PER with $\alpha = 0.6$, and NRES-controlled PER with $\tau = 0.30$ and $K = 500$. Each method runs for 1,000,000 frames using five matched seeds; task-tuned PER and ReaPER are evaluated only in the low-dimensional environments.

2.2. Normalized replay effective support and adaptive control

For TD error δ_i and positive offset ε, PER assigns priorities and sampling probabilities [1]:

$$p_i = |\delta_i| + \varepsilon, \qquad q_i(\alpha) = \frac{p_i^{\alpha}}{\sum_{j=1}^{N} p_j^{\alpha}}. \tag{1}$$

where N is the current buffer size. Uniform replay corresponds to $\alpha = 0$. Importance weights are proportional to $(Nq_i)^{-\beta}$. PER and NRES-controlled PER normalize these weights by the minibatch maximum; the official-aligned ReaPER implementation retains its buffer-wide normalization.

NRES is defined by Eq. (2), which normalizes the inverse second moment of the replay probabilities by the current buffer size [8].

$$\mathrm{NRES}(q) = \frac{1}{N \sum_{i=1}^{N} q_i^2}. \tag{2}$$

It lies in $\left[\frac{1}{N}, 1\right]$, with uniform replay giving 1. A value of 0.20 corresponds to effective support of $0.20N$ transitions. This proposal-support measure is distinct from importance-weight diagnostics.

For fixed positive priorities and $\alpha \geq 0$,

$$\frac{d}{d\alpha} \log \mathrm{NRES}(\alpha) = 2\mathbb{E}_{q_\alpha}[\log p] - 2\mathbb{E}_{q_{2\alpha}}[\log p] \leq 0. \tag{3}$$

Here $q_{c\alpha}$ denotes sampling probabilities proportional to $p_i^{c\alpha}$. Equation (3) shows that NRES is non-increasing in α for fixed positive priorities because the expected log-priority is non-decreasing with the exponent. This monotonicity permits bisection over the admissible exponent interval, conceptually similar to the adaptive use of effective-sample-size criteria in sequential Monte Carlo

DOI: 10.54254/2755-2721/2026.38508

[9]. Accordingly, Eq. (4) selects the largest admissible priority exponent that satisfies the prescribed support floor.

$$\alpha_t = \max\{\alpha \in [0,1] : \mathrm{NRES}_t(\alpha) \geq \tau\}. \tag{4}$$

The equality-reachable interval is

$$\mathscr{F}_t = [\mathrm{NRES}_t(1), \mathrm{NRES}_t(0)]. \tag{5}$$

If τ lies within this interval, bisection solves for α such that $|\mathrm{NRES}_t(\alpha) - \tau| \leq 10^{-5}$. If $\tau < \mathrm{NRES}_t(1)$, exact equality is infeasible; the controller therefore uses $\alpha = 1.0$, which still satisfies the support-floor constraint. Starting from $\alpha = 0.6$, control begins at step 7,500 for low-dimensional tasks and 10,000 for MinAtar, then repeats every K steps. Each event follows the network and priority updates and rebuilds the sampling tree. The floor applies at recalibration within numerical tolerance; subsequent priority changes can cause drift.

2.3. Evaluation metrics

Replay diagnostics include NRES, normalized Shannon entropy,

$$\widehat{H}(q) = -\frac{\sum_{i=1}^{N} q_i \log q_i}{\log N}, \tag{6}$$

and Top-1% and Top-10% probability mass. Higher entropy indicates more diffuse sampling; higher top-probability mass indicates stronger concentration. Realized diversity is the unique-buffer-index count divided by replay draws in a recent sampling window; low-dimensional diagnostics use 10,000 draws. MinAtar replay diagnostics occur every 1,250 frames, and reported MinAtar replay-state averages use checkpoints from frame 10,000 onward.

Post-control target error is

$$e_t^{\mathrm{post}} = \left|\mathrm{NRES}_t^{\mathrm{post}} - \tau\right|. \tag{7}$$

Errors at equality-reachable events assess numerical accuracy. Separate statistics summarize exact-target reachability and boundary saturation. Scheduled diagnostic error measures deviation from τ, while inter-control drift measures deviation from the latest post-control support. Floor satisfaction allows tolerance 10^{-4}. Controller overhead is calibration and tree-rebuilding time divided by total runtime.

For final performance comparisons, greedy evaluation uses five episodes every 10,000 steps in the low-dimensional environments or every 50,000 frames in MinAtar, with twenty episodes used at the final checkpoint. Table 1 and Table 2 report mean ± sample standard deviation across seeds, whereas Figures 2 and 3 show across-seed mean learning curves with shading indicating ±1 SEM. The curves use the recorded evaluation checkpoints without smoothing. Time-normalized AUC is computed using the trapezoidal integral of checkpoint returns divided by the interval between the first and last checkpoints, excluding the period before the first evaluation. Comparisons use seed-level paired differences in final return and AUC. The analysis estimates 95% percentile bootstrap confidence intervals for mean paired differences using 100,000 paired resamples.

DOI: 10.54254/2755-2721/2026.38508

3. Results and discussion

3.1. Replay concentration and support regulation

Figure 1 illustrates replay-support variation under fixed prioritization in the low-dimensional environments. In Figure 1(a–c), lines show ten-seed mean NRES with ±1 SEM, while Figure 1(d–f) summarizes controller dynamics at $\tau = 0.20$ for different control intervals. From step 7,500 onward, fixed PER with $\alpha = 1.0$ produced substantially lower mean NRES than $\alpha = 0.6$ across all three tasks, confirming stronger replay concentration. Complementary diagnostics described related but distinct properties: under $\alpha = 1.0$, the top 1% of transitions received 13.9–19.6% of replay probability, normalized entropy remained relatively high at 0.877–0.889, and recent unique-per-draw ratios fell to 0.368–0.378 compared with approximately 0.630 under uniform replay. Thus, high entropy could coexist with substantially reduced effective support. In MinAtar, fixed PER with $\alpha = 0.6$ produced mean NRES values ranging from 0.378 in Space Invaders to 0.537 in Freeway, further showing that a common exponent does not define a common replay-support state.

NRES-controlled PER regulated this variation directly. With $\tau = 0.30$ and $K = 500$, mean MinAtar NRES remained at 0.300–0.301, while the required mean α varied from 0.669 in Space Invaders to 0.860 in Freeway. Exact equality was reachable at all recalibration events in Asterix, Freeway, and Seaquest and at more than 99% of events in Breakout and Space Invaders. At equality-reachable events, mean absolute post-control error was below 5×10^{-6} in every game, and all recalibrations satisfied the support floor within 10^{-4}. These results show that matching replay support can require substantially different priority exponents across tasks.

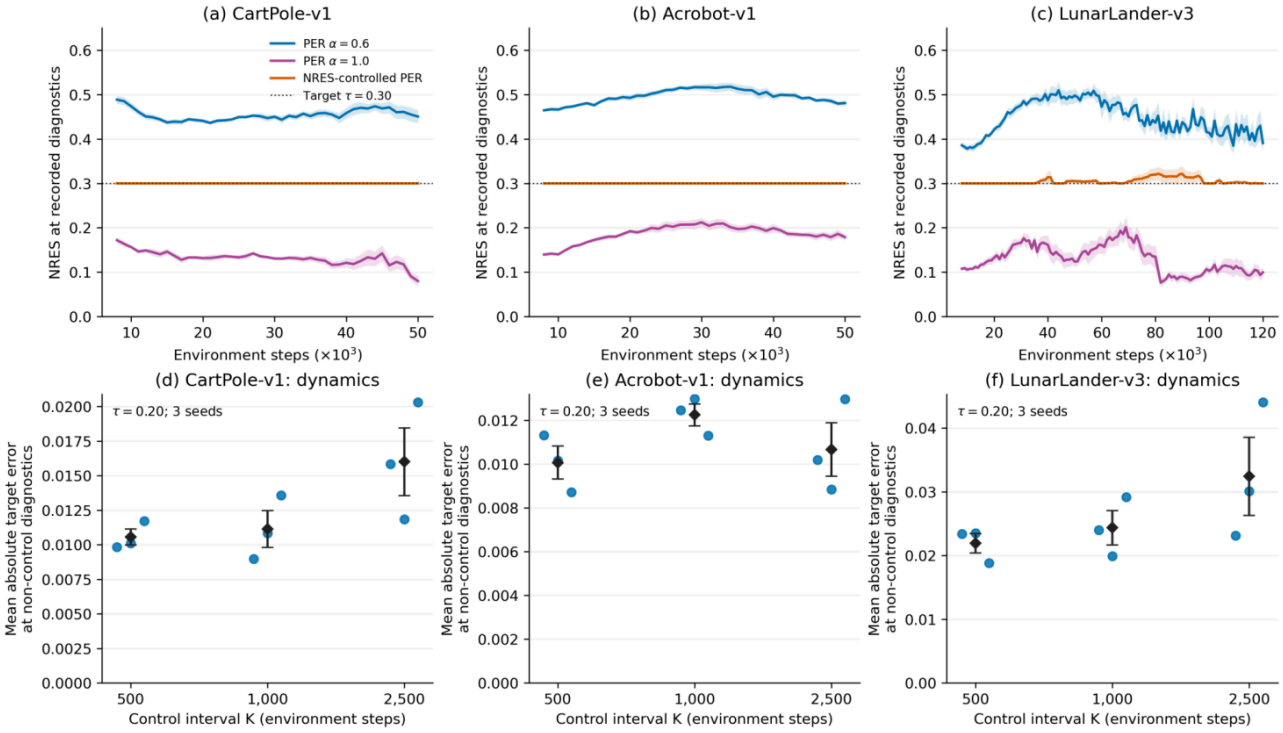

Figure 1. Replay support and controller tracking in low-dimensional environments (picture credit: original)

DOI: 10.54254/2755-2721/2026.38508

Accurate recalibration did not eliminate between-control drift. Across MinAtar, mean absolute target error between recalibrations was 0.0020–0.0039, and some diagnostic NRES values temporarily fell below the target. In the LunarLander controller-dynamics experiment, reducing K from 2,500 to 500 reduced mean absolute non-control target error from 0.0324 to 0.0219. Calibration and tree rebuilding accounted for 1.47–2.91% of total runtime across the five MinAtar games. Overall, adaptive control maintained a common replay-support target with small numerical error and limited overhead, while the remaining drift reflected changes in priorities between recalibrations.

3.2. Learning performance

Learning effects were strongly environment dependent (Figure 2). In CartPole, NRES-controlled PER achieved competitive final return and AUC, but all paired 95% bootstrap intervals for NRES-minus-baseline differences included zero, so no reliable advantage was established. In Acrobot, NRES obtained a final return comparable with the stronger prioritized baselines, but its AUC was lower than both tuned PER and ReaPER. LunarLander provided the clearest unfavorable case: uniform replay achieved substantially higher final return and AUC than NRES. These results show that successful support regulation did not consistently improve learning in the low-dimensional tasks.

Table 1. Final return and time-normalized AUC in low-dimensional environments

Environment	Method	Seeds	Final return	Time-normalized AUC
CartPole-v1	Uniform	10	219.47 ± 168.01	181.60 ± 52.73
	Tuned PER	10	229.42 ± 130.06	178.70 ± 55.48
	ReaPER	10	343.50 ± 118.15	205.48 ± 35.20
	NRES	10	323.80 ± 152.02	207.56 ± 48.27
Acrobot-v1	Uniform	10	−356.26 ± 190.87	−439.90 ± 69.26
	Tuned PER	10	−267.67 ± 158.25	−298.75 ± 82.45
	ReaPER	10	−222.27 ± 147.78	−298.61 ± 57.59
	NRES	10	−259.43 ± 184.05	−385.55 ± 58.09
LunarLander-v3	Uniform	10	139.89 ± 95.16	−5.49 ± 41.99
	Tuned PER	10	54.24 ± 107.50	−28.14 ± 48.61
	ReaPER	10	−11.17 ± 76.94	−72.39 ± 23.54
	NRES	10	11.74 ± 117.80	−91.26 ± 37.66

The five MinAtar games provided a more varied pattern (Figure 3). In Asterix, NRES achieved 26.35 final return and 17.28 AUC, compared with 16.45/8.26 for uniform replay and 23.27/17.11 for fixed PER. The paired NRES-minus-uniform intervals excluded zero for both final return and AUC, whereas comparisons with fixed PER did not. Freeway showed little difference in final return, but NRES achieved an AUC of 39.21 compared with 29.92 for uniform replay and 35.08 for fixed PER; paired AUC intervals excluded zero against both baselines. Seaquest produced the strongest positive result: NRES reached 28.58 final return and 7.44 AUC, compared with 3.87/1.19 for uniform replay and 12.89/3.55 for fixed PER. Paired intervals excluded zero for both measures against both baselines.

DOI: 10.54254/2755-2721/2026.38508

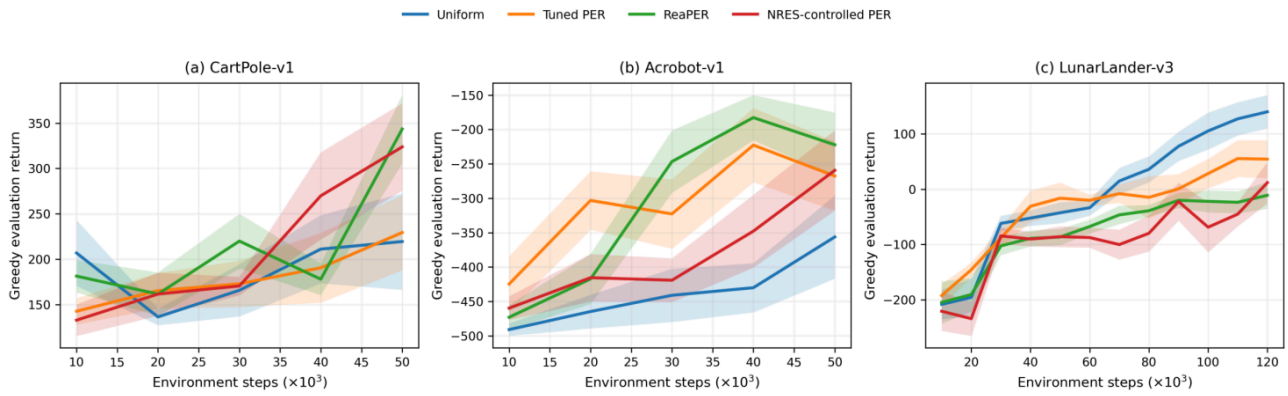

Figure 2. Learning curves in low-dimensional environments (picture credit: original)

Table 2. Final return and time-normalized AUC in MinAtar environments

Environment	Method	Seeds	Final return	Time-normalized AUC
Asterix	Uniform	5	16.45 ± 5.15	8.26 ± 1.68
	Fixed PER	5	23.27 ± 7.52	17.11 ± 2.93
	NRES	5	26.35 ± 4.37	17.28 ± 3.35
Breakout	Uniform	5	17.04 ± 3.39	11.04 ± 0.75
	Fixed PER	5	15.52 ± 5.87	11.65 ± 0.46
	NRES	5	15.35 ± 2.53	12.07 ± 1.57
Freeway	Uniform	5	54.60 ± 3.57	29.92 ± 10.37
	Fixed PER	5	53.27 ± 2.45	35.08 ± 5.66
	NRES	5	55.01 ± 3.55	39.21 ± 5.37
Seaquest	Uniform	5	3.87 ± 1.22	1.19 ± 0.12
	Fixed PER	5	12.89 ± 6.83	3.55 ± 1.87
	NRES	5	28.58 ± 8.43	7.44 ± 2.11
Space Invaders	Uniform	5	42.67 ± 11.01	38.35 ± 6.34
	Fixed PER	5	48.28 ± 12.72	38.07 ± 3.24
	NRES	5	39.90 ± 14.80	38.40 ± 1.30

Taken together, the learning results were strongly task- and comparator-dependent. Seaquest provided the clearest benefit; Asterix improved over uniform replay but not reliably over fixed PER, while Freeway showed an AUC advantage without a stable final-return gain. CartPole remained inconclusive, and Breakout and Space Invaders showed no stable benefit. Acrobot and LunarLander contained unfavorable comparisons against stronger baselines. Thus, accurate control of replay support did not by itself guarantee improved learning performance. Replay support characterizes how broadly replay probability mass is distributed, but not whether the emphasized transitions provide useful or reliable learning signals.

DOI: 10.54254/2755-2721/2026.38508

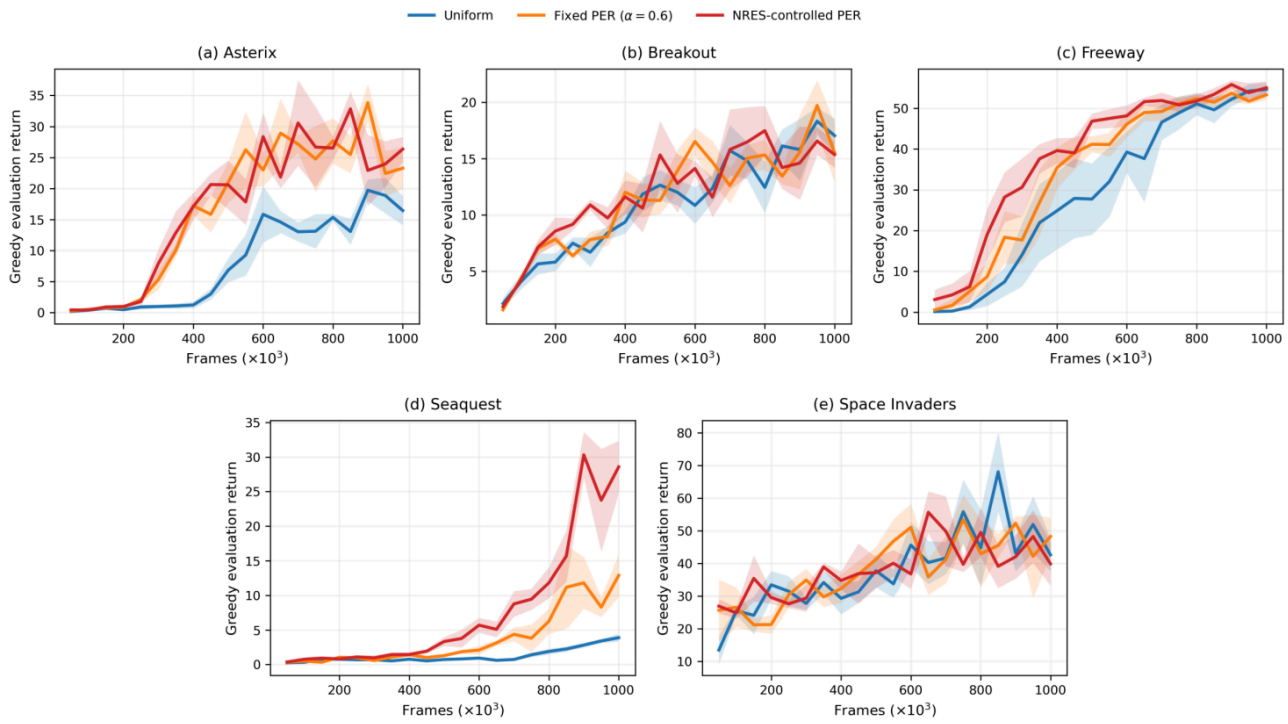

Figure 3. Learning curves across five MinAtar games (picture credit: original)

3.3. Limitations

The main limitations are the five-seed MinAtar evaluation, substantial return variability, finite training horizons, and the use of a single preselected support target in the final comparisons. In addition, task-tuned PER and ReaPER were evaluated only in the low-dimensional environments. Broader benchmark coverage, larger seed sets, and independently selected support targets are needed to determine when controlled replay support improves learning.

4. Conclusion

This study examined replay concentration in Prioritized Experience Replay through Normalized Replay Effective Support, a normalized effective-support measure derived from replay probabilities. The same priority exponent produced different levels of replay support across tasks. Adjusting the exponent allowed support to meet a prescribed floor at recalibration, with small numerical error and limited computational overhead. The effects on learning were less consistent. Some MinAtar tasks benefited, particularly Seaquest, while other environments showed limited or unfavorable outcomes. Replay support helps describe and regulate how sampling probability is distributed, but it does not establish whether the transitions being emphasized provide useful or reliable learning signals. The evaluation used five seeds in MinAtar, finite training horizons, and one preselected support target. Further experiments should test a wider set of benchmarks and task-dependent support targets, and examine whether combining support control with measures of transition usefulness or reliability improves learning.

DOI: 10.54254/2755-2721/2026.38508

References

[1] Schaul, T., Quan, J., Antonoglou, I., & Silver, D. (2015). Prioritized experience replay. arXiv preprint arXiv:1511.05952.

[2] Pan, Y., Mei, J., Farahmand, A. M., White, M., Yao, H., Rohani, M., & Luo, J. (2022, August). Understanding and mitigating the limitations of prioritized experience replay. In *Uncertainty in Artificial Intelligence* (pp. 1561-1571). PMLR.

[3] Lahire, T., Geist, M., & Rachelson, E. (2021). Large batch experience replay. arXiv preprint arXiv:2110.01528.

[4] Sujit, S., Nath, S., Braga, P., & Ebrahimi Kahou, S. (2023). Prioritizing samples in reinforcement learning with reducible loss. *Advances in Neural Information Processing Systems, 36*, 23237-23258.

[5] Li, H., Qian, X., & Song, W. (2024). Prioritized experience replay based on dynamics priority. *Scientific Reports, 14*(1), 6014.

[6] Pleiss, L. S., Sutter, T., & Schiffer, M. (2026, April). Reliability-adjusted prioritized experience replay. In *International Conference on Learning Representations (Vol. 2026)*, 59206-59232.

[7] Korniak, M., Czarnecki, M., As, Y., Miłoś, P., Abbeel, P., & Nauman, M. (2026). When does non-uniform replay matter in reinforcement learning? arXiv preprint arXiv:2605.10236.

[8] Hill, M. O. (1973). Diversity and evenness: A unifying notation and its consequences. *Ecology, 54*(2), 427–432.

[9] Beskos, A., Jasra, A., Kantas, N., & Thiery, A. (2016). On the convergence of adaptive sequential Monte Carlo methods.

[10] Van Hasselt, H., Guez, A., & Silver, D. (2016, March). Deep reinforcement learning with double q-learning. In *Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 30, No. 1)*.

[11] Young, K., & Tian, T. (2019). Minatar: An atari-inspired testbed for thorough and reproducible reinforcement learning experiments. arXiv preprint arXiv:1903.03176.