# Efficient Diversity-based Experience Replay for Deep Reinforcement Learning

**Kaiyan Zhao[1], Yiming Wang[2], Yuyang Chen[3], Yan Li[4], Leong Hou U[2], Xiaoguang Niu[1]**
[1]School of Computer Science, Wuhan University, Wuhan, China
[2]State Key Laboratory of Internet of Things for Smart City, University of Macau, Macao, China
[3]School of Professional Education, Northwestern University, USA
[4]School of Artificial Intelligence, Shenzhen Polytechnic University, China
{zhao.kaiyan, xgniu}@whu.edu.cn,wang.yiming@connect.um.edu.mo,yb57411@szpu.edu.cn
chenyuyang0520@gmail.com,ryanlhu@um.edu.mo

## Abstract

Experience replay is widely used to improve learning efficiency in reinforcement learning by leveraging past experiences. However, existing experience replay methods, whether based on uniform or prioritized sampling, often suffer from *low efficiency*, particularly in real-world scenarios with *high-dimensional state spaces*. To address this limitation, we propose a novel approach, Efficient Diversity-based Experience Replay (EDER). EDER employs a determinantal point process to model the diversity between samples and prioritizes replay based on the diversity between samples. To further enhance learning efficiency, we incorporate Cholesky decomposition for handling large state spaces in realistic environments. Additionally, rejection sampling is applied to select samples with higher diversity, thereby improving overall learning efficacy. Extensive experiments are conducted on robotic manipulation tasks in MuJoCo, Atari games, and realistic indoor environments in Habitat. The results demonstrate that our approach not only significantly improves learning efficiency but also achieves superior performance in high-dimensional, realistic environments.

## 1 Introduction

In recent years, Deep Reinforcement Learning [François-Lavet *et al.*, 2018; Wang *et al.*, 2024a] has surged in popularity, achieving remarkable success in complex decision-making tasks. DRL has been successfully applied to games [Schrittwieser *et al.*, 2020; Silver *et al.*, 2017], robotic control [Andrychowicz *et al.*, 2020; Levine *et al.*, 2016], autonomous driving scenarios including traffic light control [Yang *et al.*, 2023b; Yang *et al.*, 2023a; Yang *et al.*, 2024], and other domains, demonstrating its powerful learning and decision-making capabilities.

However, DRL still faces significant challenges in practical applications, particularly in handling sparse reward signals [Hare, 2019], high-dimensional state spaces [Ibrahimi *et al.*, 2012], and low sample efficiency [Yarats *et al.*, 2021; Wang *et al.*, 2024b]. Sparse reward signals make it difficult for agents to learn effective policies from limited positive feedback, resulting in slow and inefficient learning processes. Additionally, high-dimensional state spaces further complicate the learning process and increase computational burdens, making existing methods inefficient in large-scale and complex environments.

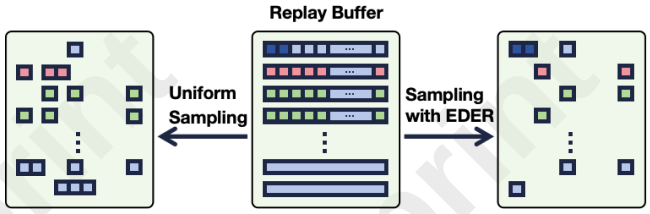


Figure 1: Sample distribution comparison of the replay buffer. Left: Uniform sampling results in an imbalanced distribution, with some data types overrepresented and others underrepresented. Right: Our method achieves a more balanced and diverse selection of samples, enhancing overall diversity and improving learning efficiency.

To address these issues, Experience Replay (ER) has been widely adopted as a key mechanism. ER improves sample efficiency and stabilizes the learning process by storing the agents' past experiences and randomly sampling them for training. Despite the improvements ER offers in sample efficiency, existing methods still suffer from inefficiency and suboptimal performance in high-dimensional state spaces. Recent studies [Andrychowicz *et al.*, 2020; Levine *et al.*, 2016; Todorov *et al.*, 2012; Jiang *et al.*, 2024; Zhao and Tresp, 2018; Fang *et al.*, 2019] have focused on enhancing ER's sampling strategies to improve their applicability and efficiency in complex environments. For instance, Hindsight Experience Replay (HER) [Andrychowicz *et al.*, 2017] generates more positive feedback samples to enhance learning efficiency; Prioritized Experience Replay (PER) [Schaul *et al.*, 2015] assigns priorities to samples based on their temporal difference (TD) errors; and Topological Experience Replay (TER) [Hong *et al.*, 2022] builds a trajectory graph and performs breadth-first updates from terminal states. Large Batch Experience Replay (LaBER) [Lahire *et al.*, 2022] improves sample efficiency by sampling large batches and performing focused updates. The Reducible Loss (ReLo) method [Sujit *et al.*, 2023] ranks samples based on their learnability, measured by consistent loss reduction. However, these approaches generally struggle to efficiently select

valuable samples in high-dimensional state spaces, leading to persistent issues of low efficiency and high-dimensional state space challenges in DRL.

To tackle these challenges, we propose a novel Experience Replay framework, Efficient Diversity-based Experience Replay (EDER). EDER utilizes Determinantal Point Processes (DPP) [Kulesza *et al.*, 2012] to model the diversity among samples and determines replay priorities based on this diversity, effectively avoiding the redundant sampling of ineffective data points. Furthermore, to handle high-dimensional state spaces in real-world environments, EDER employs Cholesky decomposition [Krishnamoorthy and Menon, 2013], significantly reducing computational complexity. Combined with rejection sampling techniques [Neal, 2003; Azadi *et al.*, 2018], EDER selects samples with higher diversity for training, thereby further enhancing overall learning efficiency.

Our main contributions are as follows. Firstly, we propose the Efficient Diversity-based Experience Replay (EDER) framework, which prioritizes sample diversity and significantly enhances experience replay (ER) efficiency, especially in high-dimensional state spaces and environments with sparse rewards. Secondly, we introduce Cholesky decomposition and rejection sampling to effectively address computational bottlenecks in large state spaces and optimize the ER mechanism by selecting more diverse samples. Lastly, we conduct extensive experimental validations across multiple complex environments, including Habitat [Savva *et al.*, 2019], Atari games [Mnih, 2013], and MuJoCo [Todorov *et al.*, 2012]. The results demonstrate that EDER not only significantly improves learning efficiency but also achieves superior performance in high-dimensional, realistic environments, thereby validating its effectiveness and adaptability in various complex settings.

## 2 Preliminaries

**Reinforcement Learning.** Reinforcement Learning (RL) is a learning paradigm where agents autonomously learn to make sequential decisions by interacting with an environment, with the goal of maximizing cumulative rewards. The problem is typically formalized as a Markov Decision Process (MDP), which is defined by a tuple $\langle S, A, P, R, \gamma \rangle$, where $S$ represents the state space, $A$ represents the action space, $P$ defines the state transition probabilities, $R$ denotes the reward function, and $\gamma$ is the discount factor. At each discrete time step $t$, the environment is in a state $s_t$, and the agent selects an action $a_t$ according to a policy $\pi$. The environment then transitions to a new state $s_{t+1}$ based on the transition probability $P(s_{t+1} \mid s_t, a_t)$, and the agent receives a scalar reward $r_{t+1}$. The agent's objective is to learn an optimal policy $\pi^*$ that maximizes the expected cumulative discounted reward starting from any initial state $s_t$:

$$V^{\pi}(s_t) = \mathbb{E}\left[\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \mid s_t = s, \pi\right],$$

where $V^{\pi}(s_t)$ is the value function that estimates the expected return when following policy $\pi$ from state $s_t$.

**Experience Replay.** Experience replay is essential in deep reinforcement learning, enabling agents to store and revisit past experiences via a replay buffer. This mechanism mitigates the issue of correlated data in online learning and improves sample efficiency. Two prominent techniques that enhance experience replay are Prioritized Experience Replay (PER) and Hindsight Experience Replay (HER): PER improves replay efficiency by prioritizing experiences based on their learning value, typically measured by the temporal difference (TD) error $\delta_t = r_{t+1} + \gamma V(s_{t+1}) - V(s_t)$, where $\gamma$ is the discount factor and $V(s_t)$ is the value function of state $s_t$.In PER, an experience is assigned a priority $p_t = |\delta_t| + \epsilon$, where $\epsilon$ ensures non-zero priority. The probability $P(i)$ of sampling an experience is proportional to its priority:

$$P(i) = \frac{p_i^{\alpha}}{\sum_k p_k^{\alpha}},$$

where $\alpha$ controls the degree of prioritization. By focusing on experiences with higher TD errors, PER enhances learning efficiency and accelerates convergence. HER addresses the challenge of sparse rewards by augmenting the replay buffer with re-labeled experiences, wherein failed attempts are reinterpreted as successes for alternative goals. Specifically, if the agent fails to achieve the intended goal $g$ at state $s_t$, HER re-labels this experience as successful for a new goal $g'$, such as a subsequent state $s_{t+k}$. The re-labeled reward function is defined as follows:

$$r_{t+1} = \begin{cases} 1 & \text{if } s_{t+k} = g', \\ 0 & \text{otherwise.} \end{cases}$$

This approach increases the number of successful experiences, thereby enhancing learning efficiency in environments with sparse rewards by effectively increasing the density of positive samples.

**Determinantal Point Processes.** Determinantal Point Processes (DPPs) are widely used probabilistic models that capture diversity within a set of points. For a discrete set $Y = \{x_1, x_2, \ldots, x_N\}$, a DPP defines a probability measure over all possible subsets of $Y$, where the probability of selecting a subset $Y \subseteq Y$ is proportional to the determinant of a positive semi-definite kernel matrix $L$ corresponding to $Y$. Specifically, the probability of sampling a subset $Y$ is:

$$P(Y) = \frac{\det(L_Y)}{\det(L + I)},$$

where $L_Y$ is the principal submatrix of $L$ indexed by the elements in $Y$, and $I$ is the identity matrix. The determinant $\det(L_Y)$ measures the diversity of $Y$ by the volume spanned by the vectors associated with $Y$. In practice, the kernel matrix $L$ is often the Gram matrix $L = X^T X$, where each column of $X$ represents a feature vector of an element in $Y$. The geometric interpretation of DPPs implies that subsets with more orthogonal feature vectors—indicating higher diversity—are more likely to be selected. This makes DPP effective for sampling diverse trajectories and goals in reinforcement learning, where diversity in the experience buffer is crucial for robust learning.

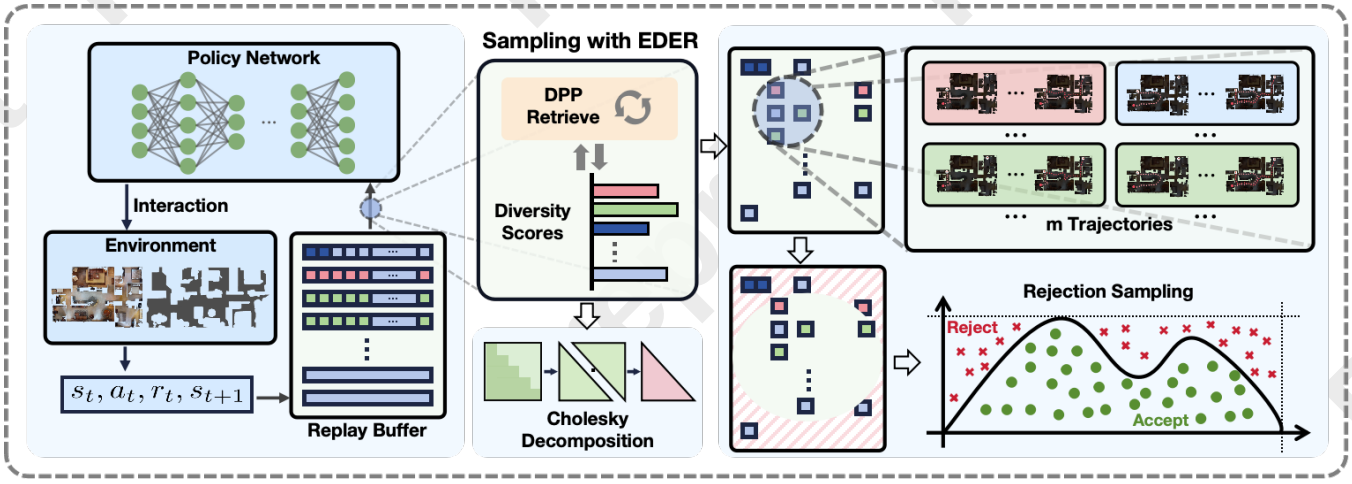


Figure 2: In the EDER framework, we leverage the Determinantal Point Process (DPP) to compute diversity scores for trajectories via Cholesky decomposition, enhancing the sampling process. Specifically, our method first uses these diversity scores to select the top $m$ most diverse trajectories. Next, we apply a rejection sampling technique to choose a subset of these trajectories for policy updates. The resulting diverse samples facilitate more efficient learning, particularly in high-dimensional environments.

## 3 Methodology

In this study, we propose a novel approach named Efficient Diversity-based Experience Replay (EDER), which enhances exploration and sample efficiency in reinforcement learning (RL) through a diversity-based trajectory selection module, which selects transitions from each trajectory based on their diversity rankings. The EDER algorithm leverages Determinantal Point Processes (DPPs) to evaluate the diversity of trajectories, enabling the exploration of a broader range of informative data. Following exploration, high-quality data is replayed to improve training efficiency. Furthermore, we employ Cholesky decomposition and rejection sampling to enhance computational efficiency, particularly in realistic environments with high-dimensional state spaces.

**Data Preprocessing.** We define the state transition dataset $T$ as a collection of state transitions accumulated during the agent's interaction with the environment, represented as: $T = \{\{s_0, s_1\}, \{s_2, s_3\}, \ldots, \{s_{T-1}, s_T\}\}$ where each element $\{s_i, s_{i+1}\}$ represents a transition from state $s_i$ to state $s_{i+1}$. In our framework, we partition $T$ into multiple partial trajectories of length $b$, denoted as $\tau_j$, each covering a state transition from $t = js$ to $t = js + b - 1$, where $s$ represents the sliding step length. The trajectories are quantified by sliding the window of length $s = b$, where the meticulous segmentation allows us to analyze and understand the behavioral patterns of intelligent agents at different stages. The specific formula is as follows:

$$T = \left\{ \{s_{jb+i}\}_{i=0}^{b-1} \mid j = 0, 1, \ldots, \frac{T}{b} - 1 \right\} \tag{1}$$

Here, $\tau_j$ denotes the partial trajectory of group $j$ covering the state transition from $s_{jb}$ to $s_{jb+b-1}$. Each $\tau_j$ is a sliding window of length $b$, demonstrating the behavior of the agent and its environmental adaptation during that time period.

### 3.1 Diversity-Based Trajectory Selection Module

The objective of this module is to select diverse trajectories from the replay buffer, enhancing learning by utilizing a wide range of experiences. A set of summary timelines describing the key trajectory events is generated from the entire collection of trajectories, which involves the following steps:

**Trajectory Segmentation.** The entire sequence of state transitions during an interaction, denoted as $\tau$, is segmented into several partial trajectories $\tau_j$ of length $b$. Each segment $\tau_j$ covers transitions from state $s_n$ to $s_{n+b-1}$, allowing for detailed capture of dynamics between state transitions. For clarity, we set a sliding window of $b = 2$ in this part, while other values are explored in the ablation studies. Under this setting, a trajectory $\tau$ can be divided into $N_p$ partial segments.

$$\tau = \{\underbrace{\{s_0, s_1\}}_{\tau_1}, \underbrace{\{s_2, s_3\}}_{\tau_2}, \underbrace{\{s_4, s_5\}}_{\tau_3}, \ldots, \underbrace{\{s_{T-1}, s_T\}}_{\tau_{N_p}}\}$$

**Diversity Assessment.**To effectively evaluate the diversity of each partial trajectory $\tau_j$, we adopt the theoretical framework of Determinantal Point Processes(DPPs). Specifically, the diversity metric $d_{\tau_j}$ for a partial trajectory $\tau_j$ is defined as the determinant of its corresponding kernel matrix:

$$d_{\tau_j} = \det(L_{\tau_j}) \tag{2}$$

Intuitively, the determinant quantifies the $n$-dimensional volume spanned by the embedded state transitions in $\tau_j$, assigning higher values to sets of transitions that are more linearly orthogonal and thus more diverse. Here, $L_{\tau_j}$ is the kernel matrix constructed from the state transitions within trajectory $\tau_j$, defined as:

$$L_{\tau_j} = M^T M \tag{3}$$

The columns of matrix $M$ are the $\ell_2$-normalized vector representations $\hat{s}$ of each state $s$ in trajectory $\tau_j$.

**Theorem 1** (Correlation between Determinant and Diversity)**.** *Let $M \in \mathbb{R}^{d\times b}$ be a matrix whose columns are the $\ell_2$-normalized state vectors $\hat{s}$ in trajectory $\tau_j$. The determinant $\det(L_{\tau_j})$ of the kernel matrix $L_{\tau_j} = M^T M$ reaches its maximum value when the state vectors are mutually orthogonal, indicating the highest diversity of the trajectory.*

Proof in Appendix A. The choice of Determinantal Point Processes is motivated by the ability of $\det(L_{\tau_j})$ to effectively measure the diversity of state vectors within trajectory $\tau_j$. Based on Theorem 1, a larger determinant indicates higher diversity of the trajectory.

The determinant $\det(L_{\tau_j}) = \det(M^T M)$ is equal to the square of the volume of the parallelepiped spanned by the columns of matrix $M$. When the vectors are mutually orthogonal, the volume and thus the determinant reaches its maximum value, reflecting the highest independence and diversity of the state vectors. Conversely, if the vectors are linearly dependent, both the volume and the determinant decrease, indicating reduced diversity. Additionally, in DPPs, the kernel matrix $L_{\tau_j}$ captures the similarities between state vectors, inherently favoring the selection of diverse and minimally redundant subsets. Therefore, DPPs are an ideal choice for evaluating the diversity of trajectories in reinforcement learning [Kunaver and Požrl, 2017]. A larger $d_{\tau_j}$ indicates that the state vectors are more uniformly distributed in the feature space with lower similarity, reflecting higher diversity. This is crucial for policy training in reinforcement learning, as diversified data facilitates better policy generalization and adaptation to various environmental conditions.

**Sampling Strategy.** The total diversity of a trajectory $\tau$, denoted as $d_\tau$, is defined as the sum of the diversities of all its constituent partial trajectories:

$$d_\tau = \sum_{j=1}^{N_p} d_{\tau_j} \tag{4}$$

Equation (4) provides a comprehensive measure, effectively reflecting the overall diversity of the trajectory. We employ a non-uniform sampling strategy to prioritize trajectories with higher diversity:

$$p(\tau_i) = \frac{d_{\tau_i}}{\sum_{n=1}^{N_e} d_{\tau_n}}, \tag{5}$$

where $N_e$ is the total number of trajectories in the replay buffer, this strategy enhances learning efficiency by increasing the likelihood of selecting highly diverse trajectories, thereby enabling the agent to effectively learn and adapt to various environmental conditions.

Although the determinant effectively measures diversity, its direct computation in high-dimensional state spaces is ***computationally intensive***, especially for large trajectory lengths $b$. Therefore, we employ Cholesky decomposition and rejection sampling to optimize computation speed in the following section. The diversity metric $d_{\tau_j}$ quantifies the independence and diversity of state vectors within trajectory $\tau_j$ using the determinant.

### 3.2 Improving Computational Efficiency

Scaling to high-dimensional environments is crucial for the applicability of deep reinforcement learning algorithms. Traditional approaches often fail due to computational inefficiency, especially when dealing with large state spaces where calculations become difficult and time-consuming. Computing Determinantal Point Processes (DPPs) in high-dimensional state spaces is computationally intensive due to the ***complexity of calculating*** large kernel matrices. This challenge is particularly acute in extensive state spaces where traditional methods struggle to maintain efficiency. To address this issue, we propose an optimized approach that integrates Cholesky decomposition and rejection sampling into our method. This approach reduces computational costs while preserving the effectiveness of DPPs with theoretical guarantees, making them applicable to complex reinforcement learning scenarios.

**Cholesky Decomposition.** To simplify the determinant calculation of the kernel matrix, a key operation in DPP, we employ Cholesky decomposition. For a window length $b$, given state vectors $\hat{s}_1, \hat{s}_2, \ldots, \hat{s}_b$, we construct the matrix $M$ as $M = [\hat{s}_1, \hat{s}_2, \ldots, \hat{s}_b]$. The kernel matrix $L_{\tau_j}$ is then formed as Equation (3). To efficiently compute the determinant of $L_{\tau_j}$, we apply Cholesky decomposition, which decomposes $L_{\tau_j}$ into a product of a lower triangular matrix $L_C$ and its transpose $L_C^T$:

$$L_{\tau_j} = L_C L_C^T \tag{6}$$

The determinant is then computed as the product of the squares of the diagonal elements of $L_C$:

$$\det(L_{\tau_j}) = \prod_{i=1}^{b} l_{ii}^2 \tag{7}$$

Here, $l_{ii}$ denotes the $i$-th diagonal element. With Cholesky decomposition, the time complexity of determinant computation for each segment is reduced from $O(n^3)$ to $O(n^2)$.

**Rejection Sampling.** Numerous trajectories are utilized in training, resulting in high sampling inefficiency. To enhance ***sampling efficiency***, we introduce rejection sampling. This method effectively filters trajectory segments *before insertion into the replay buffer*, which is particularly useful in high-dimensional state spaces where storing redundant segments incurs significant computational overhead. By prioritizing segments with higher diversity scores, rejection sampling minimizes the retention of less informative segments. Consequently, computational resources are focused on the most diverse and relevant experiences, ensuring that the replay buffer contains the most valuable transitions.

The rejection sampling process is detailed as follows: First, for each trajectory segment $\tau_j$, we compute its diversity score $Q_j$ using Equation (2) and (7):

$$Q_j = d_{\tau_j} = \det(L_{\tau_j}) = \prod_{i=1}^{b} l_{ii}^2 \tag{8}$$

Next, we determine a normalization constant $M$, defined as:

$$M = \max(Q) \tag{9}$$

This ensures that for all trajectory segments $\tau_j$, the acceptance probability $\alpha = \frac{Q_j}{M}$ remains in the valid range $[0, 1]$. During the rejection sampling process, we uniformly select a candidate segment $\tau'$ from the current batch of generated segments. Then we draw a uniformly distributed random number $u \sim U(0, 1)$. If the sampled number satisfies:

$$u \leq \alpha = \frac{d_{\tau'}}{M}, \tag{10}$$

we accept the candidate trajectory segment $\tau'$; otherwise, we reject it and resample. This process yields a set of diverse trajectory segments, which are then inserted into the replay buffer. Subsequently, training batches are sampled from the buffer according to Equation (5).

### 3.3 Time Complexity Analysis

**Theorem 2.** *The time complexity of the EDER algorithm is $O(Nbd + Nb^3 + N \log m + m)$ without employing Cholesky decomposition and rejection sampling, and it is reduced to $O(Nbd + Nb^2 + N \log m + m)$ after integrating these optimizations. Here, $N$ denotes the number of state transitions, $b$ the segment length, $d$ the dimensionality of the state vectors, and $m$ the number of top trajectories selected.*

The proof is in Appendix A. Based on Theorem 2, the integration of Cholesky decomposition and rejection sampling significantly reduces the overall computational complexity of the EDER algorithm, especially when dealing with large segment lengths b. This improvement makes the EDER algorithm more efficient and scalable for high-dimensional reinforcement learning tasks, enhancing its applicability in complex environments.

**Algorithm 1** EDER

**Initialize:** Replay buffer $\mathcal{D}$, diversity score list $Q$, segment length $b$
**while** not converged **do**
  Initialize state $s_0$
  **for** $t = 1$ to $T$ **do**
    Select action $a_t$ via policy $\pi(s_t, \theta)$
    Execute $a_t$, observe $s_{t+1}$, receive $r_t$
    Store $(s_t, a_t, r_t, s_{t+1})$ in $\mathcal{D}$
  **end for**
  **for** each trajectory $\tau$ in $\mathcal{D}$ **do**
    Segment $\tau$ into sub-trajectories $\tau_j$
    Compute diversity score $Q_j$ using $\det(L_{\tau_j})$ via Cholesky Decomposition ▷ Equation (7)
    Append $Q_j$ to $Q$
  **end for**
  Set $M = \max(Q)$
  **for** each $Q_j$ in $Q$ **do**
    Calculate acceptance $\alpha = \frac{Q_j}{M}$
    Accept corresponding $\tau_j$ if $u \leq \alpha$, else discard
  **end for**
  Sample $\mathcal{B} \sim \mathcal{D}$ using Eq. (5)
  Update $\theta$ using $\mathcal{B}$
**end while**

## 4 Experiments

Our experiments aim to rigorously evaluate the performance of the proposed Efficient Diversity-based Experience Replay (EDER) method across multiple environments, focusing on its effectiveness compared to established baseline methods. The experiments are conducted in Mujoco, Atari, and real-life Habitat environments, each selected to highlight different aspects of EDER's capabilities. Detailed environment settings are provided in Appendix C. Details are available at https://arxiv.org/abs/2410.20487.

**Baselines.** We compare our method against the following baselines. DDPG [Lillicrap *et al.*, 2019]: a deep reinforcement learning algorithm for continuous action spaces, combining deterministic policy gradients with Q-learning. DQN [Mnih *et al.*, 2013]: a widely used algorithm for discrete action spaces, approximating the Q-value function with deep neural networks. HER [Andrychowicz *et al.*, 2017]: Hindsight Experience Replay enables learning from alternative goals that could have been achieved, improving efficiency in sparse reward settings. PER [Schaul *et al.*, 2015]: Prioritized Experience Replay enhances learning by prioritizing important transitions. TER [Hong *et al.*, 2022]: Topological Experience Replay builds a graph from experience trajectories to track predecessors, then performs breadth-first updates from terminal states like reverse topological sorting. LaBER [Lahire *et al.*, 2022]: Large Batch Experience Replay samples a large batch from the replay buffer, computes gradient norms, downsamples to a smaller batch based on priority, and uses this mini-batch to update the policy. Relo [Sujit *et al.*, 2023]: Reducible Loss (ReLo) is a sample prioritization method that ranks samples by their learnability, measured by the consistent reduction in their loss over time.

| Methods | Residential | Office | Commercial |
|---|---|---|---|
| DDPG | 9.0 ± 2.5 | 27.5 ± 1.9 | 23.0 ± 2.0 |
| DDPG+HER | 35.0 ± 2.8 | 42.5 ± 2.1 | 42.0 ± 2.3 |
| DDPG+PER | 23.0 ± 3.0 | 45.0 ± 2.3 | 34.5 ± 2.4 |
| DDPG+TER | 48.0 ± 2.1 | 47.0 ± 1.5 | 54.5 ± 3.5 |
| DDPG+LaBER | 52.0 ± 3.0 | 54.0 ± 2.3 | 48.5 ± 2.1 |
| DDPG+Relo | 55.0 ± 1.3 | 53.0 ± 2.9 | 59.5 ± 1.8 |
| DDPG+EDER w/o R.S. | 58.0 ± 3.1 | 50.8 ± 2.1 | 55.5 ± 3.3 |
| DDPG+EDER w/o C.D. | 60.1 ± 2.6 | 68.0 ± 3.9 | 58.5 ± 2.7 |
| DDPG+EDER | **64.0 ± 3.3** | **74.5 ± 2.4** | **68.4 ± 2.5** |

Table 1: Success rates (%) across environments in HM3D.

### 4.1 High-dimensional Environment

We utilize the AI Habitat platform to evaluate EDER's scalability and effectiveness in vision-based navigation tasks. Specifically, the agent is randomly initialized in the environment and relies solely on its sensory inputs for navigation. With no prior knowledge of the environment map, the agent must autonomously explore the scene and locate the target object. The success metric is defined as whether the agent successfully reaches the target object. These tasks are conducted in photorealistic 3D environments, where the high-dimensional observation space poses significant challenges for efficient exploration. We evaluate EDER in three environments from the Habitat-Matterport 3D Research Dataset

| Method | Alien | Asterix | BeamR. | Breakout | CrazyCli. | Demo. | H.E.R.O. | Krull | KungFu. | MsPac. |
|---|---|---|---|---|---|---|---|---|---|---|
| Random | 227.8 | 210.0 | 363.9 | 1.7 | 10,780 | 152.1 | 1,027.0 | 1,598.3 | 258.5 | 307.3 |
| DQN | 3,069.0 | 6,012.0 | 6,846.0 | 401.2 | 14,103.3 | 9,711.0 | 19,950.3 | 3,805.2 | 23,270.3 | 2,311.0 |
| DQN+PER | 4,204.2 | 31,527.3 | 23,384.0 | 373.9 | 141,161.0 | 71,846.7 | 23,038.1 | 9,728.6 | 39,581.2 | 6,519.1 |
| DQN+TER | 4,298.5 | 24,798.5 | 24,432.1 | 420.3 | 142,321.5 | 73,346.2 | 21,543.0 | 9,643.1 | 39,832.9 | 6,587.0 |
| DQN+LaBER | 4,365.2 | 39,172.1 | 23,543.4 | 462.2 | 145,672.2 | 75,128.0 | 24,495.0 | 9,764.7 | 41,823.0 | 6,691.4 |
| DQN+Relo | 4,312.9 | 38,432.4 | 26,064.0 | 492.5 | 144,875.0 | 75,442.1 | **26,535.3** | 9,734.4 | 41,232.0 | 6,613.1 |
| DQN+EDER w/o R.S. | 4,292.9 | 44,823.9 | 25,032.0 | 438.8 | 140,274.0 | 74,924.4 | 24,264.3 | 9,374.4 | 40,387.0 | 6,124.1 |
| DQN+EDER w/o C.D. | 4,689.4 | 50,283.7 | 25,731.0 | 481.9 | 142,328.6 | 75,326.1 | 25,214.8 | 9,353.4 | 40,983.0 | 6,493.1 |
| DQN+EDER | **4,723.1** | **54,328.5** | **26,543.0** | **516.0** | **147,305.0** | **76,150.1** | 26,246.0 | **9,805.0** | **43,310.0** | **6,722.2** |

| Method | Enduro | Freew. | Frost. | Hem | Jamesb. | Kangar. | Pong | Qbert | River. | ZaxxPH. |
|---|---|---|---|---|---|---|---|---|---|---|
| Random | 0.0 | 0.0 | 65.2 | 1,027.0 | 29.0 | 52.0 | -20.7 | 163.9 | 1,338.5 | 32.5 |
| DQN | 301.8 | 30.3 | 328.3 | 19,950.0 | 576.7 | 6,740.0 | 18.9 | 10,596.0 | 8,316.0 | 4,977.0 |
| DQN+PER | 2,093.0 | 33.7 | 4,380.1 | 23,037.7 | 5,148.0 | 16,200.0 | 20.6 | 16,256.5 | 14,522.3 | 10,469.0 |
| DQN+TER | 2,208.0 | 35.2 | 4,721.3 | 24,332.4 | 5,032.4 | 16,632.0 | 21.0 | 17,281.3 | 19,232.5 | 10,834.0 |
| DQN+LaBER | 2,165.5 | 31.6 | 4,923.5 | 24,251.9 | 5,218.2 | 16,321.0 | 21.0 | 17,744.6 | 21,368.4 | 12,832.0 |
| DQN+Relo | 2,272.2 | 37.6 | 4,892.7 | 25,232.6 | 5,209.8 | **16,820.1** | 21.0 | 19,013.2 | 22,312.7 | 14,123.0 |
| DQN+EDER w/o R.S. | 2,138.0 | 32.0 | 5,145.4 | 24,214.2 | 5,121.0 | 16,054.2 | 21.0 | 18,421.0 | 21,833.1 | 13,233.1 |
| DQN+EDER w/o C.D. | 2,332.7 | 38.1 | 5,483.1 | 25,970.3 | 5,240.1 | 16,192.1 | 21.0 | 19,192.5 | 23,382.0 | 14,523.7 |
| DQN+EDER | **2,340.0** | **39.0** | **5553.0** | **26,246.0** | **5,275.0** | 16,644.0 | **21.0** | **19,545.0** | **24,425.0** | **14,920.0** |

Table 2: Comparison of Atari Game Scores. Best results are **bold**.

(HM3D) [Ramakrishnan *et al.*, 2021], representing complex, real-world indoor spaces. Specifically, we choose a residential setting (e.g., living rooms and bedrooms), an office environment (e.g., workspaces and corridors), and a commercial space (e.g., shopping centers), each featuring open areas and diverse visual elements. Target- or topology-based approaches (e.g., HER, TER) incorporate structured global exploration strategies in complex environments to improve efficiency, while loss- or priority-based methods (e.g., PER, LaBER, ReLo) focus on refining transition sampling based on loss or priority rules. However, whether by replacing goals or leveraging topological structures for exploration or by adjusting sampling mechanisms, these methods often neglect the importance of diverse trajectories and comprehensive exploration. In contrast, EDER emphasizes leveraging diverse trajectories to enhance exploration efficiency. As shown in Table 1, EDER consistently achieves higher success rates across all experimental settings in high-dimensional visual tasks, demonstrating its effectiveness and scalability.

## 4.2 Atari Games

The second set of experiments evaluates EDER in discrete-action environments using the Atari benchmark, renowned for its challenging exploration tasks. For instance, in Alien, the agent navigates a maze, earns points by collecting bright spots, and loses a life upon contact with monsters. We test EDER+DQN across various Atari games, comparing it against standard DQN, DQN+PER, and other replay variants. The selected games, including Kangaroo and Jamesbond, are particularly demanding in terms of exploration. As shown in Table 2, EDER achieve the best performance in 18/20 games. Other methods generally fall into two categories: those that emphasize structured global exploration and those that prioritize samples based on local rewards or loss values. While effective in certain scenarios, these methods often overlook transitions that do not immediately yield high TD errors or rewards. In contrast, EDER explicitly incorporates diverse trajectories into its replay mechanism, promoting enhanced exploration and ultimately improving overall performance.

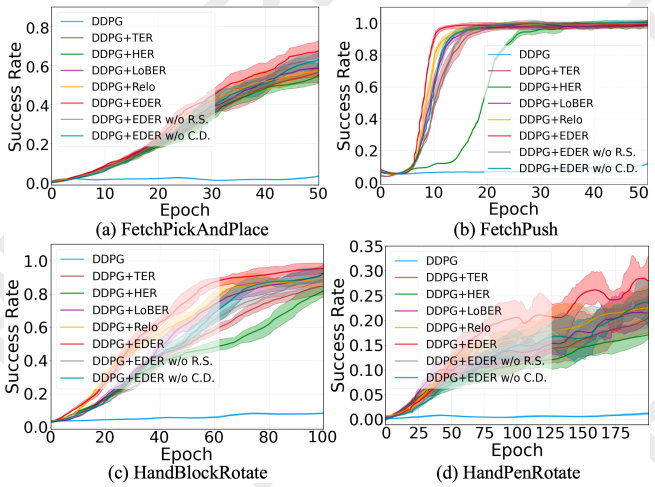


Figure 3: Success rates between EDER and other baselines

## 4.3 Mujoco Tasks

We also evaluate EDER in MuJoCo environments, focusing on continuous control tasks with sparse rewards. These tasks are particularly challenging due to their high-dimensional state and action spaces. We select four representative tasks from the FetchEnv, which involves a robotic arm with 7 degrees of freedom, and HandEnv, featuring the 24-degree-of-freedom Shadow Dexterous Hand: FetchPickAndPlace, FetchPush, HandBlockRotate, and HandPenRotate. As shown in Figure 3, EDER significantly outperforms traditional DDPG and its variants in both learning speed and success rates. Notably, EDER achieves exceptional performance in the Shadow Dexterous Hand task, demonstrating its ability

to navigate complex, high-dimensional spaces. This superior performance is attributed to EDER's capacity to enhance exploration through diverse trajectories, resulting in more efficient learning.

### 4.4 Ablation Studies

In our ablation studies, we evaluate the impact of Rejection Sampling (R.S.) and Cholesky Decomposition (C.D.) on the training efficiency of EDER. As shown in Table 3, removing R.S. reduces exploration efficiency, slows convergence, and lowers success rates. In contrast, excluding C.D. leads to increased instability and longer training times, ultimately diminishing learning efficiency. Moreover, when comparing training times with HER-based baselines, EDER maintains competitive performance, especially in more complex tasks like PickAndPlace, demonstrating favorable trade-offs between computational cost and performance. Additional experiments varying $m$ (the number of diverse trajectories) and $b$ (the trajectory length) are provided in Appendix B. Our results indicate that increasing $m$ promotes exploration by sampling more diverse trajectories, helping to avoid local optima. However, excessive $m$ increases computational burden without proportionate performance gains, leading to slower convergence. Likewise, longer segments ($b$) provide richer context and improve exploration in temporally dependent environments, but overly long trajectories may introduce instability and reduce overall training efficiency.

| **PickAndPlace Task (Training Time in Minutes)** | | | |
|---|---|---|---|
| **Method** | **Time** | **Method** | **Time** |
| DDPG + HER (Buffer) | 80.7 | DDPG + LaBER | 93.6 |
| DDPG + PER (sum-tree) | 63.4 | DDPG + Relo | 107.1 |
| DDPG + TER | 91.9 | DDPG + EDER | 103.1 |
| **Push Task: DDPG + EDER Variants (in Minutes)** | | | |
| **Method** | **Time** | **Method** | **Time** |
| EDER (b=10) | 124.3 | EDER (b=T) | 156.0 |
| w/o R.S. (b=10) | 173.2 | w/o R.S. (b=T) | 182.7 |
| w/o C.D. (b=10) | 129.2 | w/o C.D. (b=T) | 171.3 |

Table 3: Training times (in minutes) for baseline methods and EDER variants on PickAndPlace and Push tasks. R.S.: Rejection Sampling; C.D.: Cholesky Decomposition.

## 5 Related Work

The concept of Experience Replay (ER) was first introduced by [Lin, 1992], where past experiences are stored in a buffer and replayed during training to break the correlation between sequential data, which helps mitigate the non-stationarity in RL. [Mnih *et al.*, 2013] later incorporated ER into the Deep Q-Network (DQN), where the use of randomly sampled batches from the replay buffer was crucial in stabilizing the learning process and led to significant advancements in the performance of RL algorithms. Prioritized Experience Replay (PER) [Schaul *et al.*, 2015] enhances learning by focusing on high TD-error samples and prioritizing informative experiences. Various extensions to PER have been proposed, such as the actor-critic-based PER [Saglam *et al.*, 2022], which dynamically adjusts sampling priorities to balance exploration and exploitation; Attentive PER [Sun *et al.*, 2020] uses attention mechanisms to replay experiences relevant to the current learning phase, enhancing training efficiency. Additionally, recent studies have introduced new priority criteria to enhance PER's effectiveness. Relo [Sujit *et al.*, 2023] define the learnability of transitions as a priority criterion, prioritizing samples that consistently reduce training loss.TER [Hong *et al.*, 2022] builds a trajectory graph and prioritizes updates breadth-first from terminal states; LaBER [Lahire *et al.*, 2022] enhances efficiency by leveraging large batch sampling with focused updates. FSER [Yu *et al.*, 2024] combines frequency and similarity indices to prioritize rare and policy-aligned experiences. [Wei *et al.*, 2021] integrates transition revisit frequency with TD error for more effective replay buffer prioritization. Hindsight Experience Replay (HER) [Andrychowicz *et al.*, 2017], offers a novel approach to handling sparse rewards by retrospectively altering the goals of unsuccessful episodes, thereby converting failures into valuable learning experiences. HER has been integrated with techniques such as curriculum learning [Fang *et al.*, 2019] and multi-goal learning [Zhou *et al.*, 2019] to enhance the generalization and adaptability of RL agents. Additionally, Contact Energy Based Prioritization (CEBP) [Sayar *et al.*, 2024] prioritizes replay samples based on contact-rich interactions, selecting the most informative experiences. Distributed ER architectures like Ape-X [Horgan *et al.*, 2018] and IMPALA [Espeholt *et al.*, 2018] have scaled experience replay across multiple actors, significantly accelerating training while maintaining efficiency. Relay Hindsight Experience Replay [Luo *et al.*, 2023] decomposes tasks and employs a multi-goal network for self-guided exploration. Hybrid approaches have also been explored, such as combining PER and HER [Zhang *et al.*, 2017], as well as introducing adaptive replay strategies [Peng *et al.*, 2019], adjusting priorities based on learning progress. These advancements enhance the robustness and scalability of experience replay methods, enabling more efficient and effective learning across a range of reinforcement learning tasks.

## 6 Conclusion

In this work, we present the Efficient Diversity-based Experience Replay (EDER) framework, which prioritizes sample diversity to significantly enhance the efficiency of experience replay (ER), particularly in high-dimensional state spaces and environments with sparse rewards. To address computational bottlenecks in large state spaces, we integrate Cholesky decomposition and rejection sampling, enabling the selection of more diverse and representative samples while optimizing the ER mechanism. Extensive experiments on MuJoCo, Atari games, and Habitat demonstrate the superiority of EDER compared to existing approaches. EDER not only substantially improves learning efficiency but also delivers superior performance in high-dimensional and realistic environments. These results validate the effectiveness and adaptability of EDER across a variety of complex settings.

## Contribution Statement

Kaiyan Zhao and Yiming Wang contributed equally to this work, serving as the co-first author. Corresponding author Xiaoguang Niu supervised. Other co-authors contributed manuscript revision.

## Acknowledgments

This work was supported in part by the Key Research and Development Project of Hubei Province (2022BCA057), the Science and Technology Development Fund Macau SAR (0003/2023/RIC, 0052/2023/RIA1, 0031/2022/A, 001/2024/SKL for SKL-IOTSC), the Shenzhen-Hong Kong-Macau Science and Technology Program Category C (SGDX20230821095159012), the National Natural Science Foundation of China (62402325), and the Research Foundation of Shenzhen Polytechnic University (6022310014K, 6022312054K). Work partially performed on the supercomputing system at the Supercomputing Center of Wuhan University and at SICC (supported by SKL-IOTSC, University of Macau).

## References

[Andrychowicz *et al.*, 2017] Marcin Andrychowicz, Filip Wolski, Alex Ray, Jonas Schneider, Rachel Fong, Peter Welinder, Bob McGrew, Joshua Tobin, Pieter Abbeel, and Wojciech Zaremba. Hindsight experience replay. In *Neural Information Processing Systems*, 2017.

[Andrychowicz *et al.*, 2020] Marcin Andrychowicz, Bob Baker, Marcin Chociej, Rafal Jozefowicz, Brian McGrew, Jakub Pachocki, Andrew Petron, Matthias Plappert, Geoffrey Powell, Alex Ray, et al. Learning dexterous in-hand manipulation. *International Journal of Robotics Research*, 2020.

[Azadi *et al.*, 2018] Samaneh Azadi, Catherine Olsson, Trevor Darrell, Ian Goodfellow, and Augustus Odena. Discriminator rejection sampling. *arXiv preprint arXiv:1810.06758*, 2018.

[Espeholt *et al.*, 2018] Lasse Espeholt, Hubert Soyer, Remi Munos, Karen Simonyan, Volodymyr Mnih, Tom Ward, and Martin Riedmiller. Impala: Scalable distributed deep-rl with importance weighted actor-learner architectures. In *Proceedings of the 35th International Conference on Machine Learning*, pages 1407–1416. PMLR, 2018.

[Fang *et al.*, 2019] Minshu Fang, Tao Zhou, Yifeng Du, Lei Han, and Zhe Zhang. Curriculum-guided hindsight experience replay. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

[François-Lavet *et al.*, 2018] Vincent François-Lavet, Peter Henderson, Riashat Islam, Marc G Bellemare, Joelle Pineau, et al. An introduction to deep reinforcement learning. *Foundations and Trends® in Machine Learning*, 11(3-4):219–354, 2018.

[Hare, 2019] Joshua Hare. Dealing with sparse rewards in reinforcement learning. *arXiv preprint arXiv:1910.09281*, 2019.

[Hong *et al.*, 2022] Zhang-Wei Hong, Tao Chen, Yen-Chen Lin, Joni Pajarinen, and Pulkit Agrawal. Topological experience replay. *arXiv preprint arXiv:2203.15845*, 2022.

[Horgan *et al.*, 2018] Dan Horgan, John Quan, David Budden, Gabriel Barth-Maron, Matteo Hessel, Hado Van Hasselt, and David Silver. Distributed prioritized experience replay. *arXiv preprint arXiv:1803.00933*, 2018.

[Ibrahimi *et al.*, 2012] Morteza Ibrahimi, Adel Javanmard, and Benjamin Roy. Efficient reinforcement learning for high dimensional linear quadratic systems. *Advances in Neural Information Processing Systems*, 25, 2012.

[Jiang *et al.*, 2024] Yiding Jiang, J Zico Kolter, and Roberta Raileanu. On the importance of exploration for generalization in reinforcement learning. In *Advances in Neural Information Processing Systems 36*, 2024.

[Krishnamoorthy and Menon, 2013] Aravindh Krishnamoorthy and Deepak Menon. Matrix inversion using cholesky decomposition, 2013.

[Kulesza *et al.*, 2012] Alex Kulesza, Ben Taskar, et al. Determinantal point processes for machine learning. *Foundations and Trends® in Machine Learning*, 2012.

[Kunaver and Požrl, 2017] Matevž Kunaver and Tomaž Požrl. Diversity in recommender systems–a survey. *Knowledge-based systems*, 123:154–162, 2017.

[Lahire *et al.*, 2022] Thibault Lahire, Matthieu Geist, and Emmanuel Rachelson. Large batch experience replay. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 11790–11813. PMLR, 17–23 Jul 2022.

[Levine *et al.*, 2016] Sergey Levine, Chelsea Finn, Trevor Darrell, and Pieter Abbeel. End-to-end training of deep visuomotor policies. *Journal of Machine Learning Research*, 2016.

[Lillicrap *et al.*, 2019] Timothy P. Lillicrap, Jonathan J. Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning, 2019.

[Lin, 1992] Long-Ji Lin. Self-improving reactive agents based on reinforcement learning, planning and teaching. *Machine Learning*, 8(3-4):293–321, 1992.

[Luo *et al.*, 2023] Yongle Luo, Yuxin Wang, Kun Dong, Qiang Zhang, Erkang Cheng, Zhiyong Sun, and Bo Song. Relay hindsight experience replay: Self-guided continual reinforcement learning for sequential object manipulation tasks with sparse rewards. *Neurocomputing*, page 126620, 2023.

[Mnih *et al.*, 2013] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning, 2013.

[Mnih, 2013] Volodymyr Mnih. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

[Neal, 2003] Radford M Neal. Slice sampling. *The annals of statistics*, 31(3):705–767, 2003.

[Peng *et al.*, 2019] Xue Bin Peng, Aviral Kumar, Grace Zhang, and Sergey Levine. Advantage-weighted regression: Simple and scalable off-policy reinforcement learning, 2019.

[Ramakrishnan *et al.*, 2021] Santhosh K. Ramakrishnan, Aaron Gokaslan, Erik Wijmans, Oleksandr Maksymets, Alex Clegg, John Turner, Eric Undersander, Wojciech Galuba, Andrew Westbury, Angel X. Chang, Manolis Savva, Yili Zhao, and Dhruv Batra. Habitat-matterport 3d dataset (hm3d): 1000 large-scale 3d environments for embodied ai, 2021.

[Saglam *et al.*, 2022] Berkay Saglam, Fatih Burak Mutlu, Dilek Cetin Cicek, et al. Actor prioritized experience replay. *arXiv preprint arXiv:2209.00532*, 2022.

[Savva *et al.*, 2019] Manolis Savva, Abhishek Kadian, Oleksandr Maksymets, Yili Zhao, Erik Wijmans, Bhavana Jain, Julian Straub, Jia Liu, Vladlen Koltun, Jitendra Malik, Devi Parikh, and Dhruv Batra. Habitat: A platform for embodied ai research. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019.

[Sayar *et al.*, 2024] Erdi Sayar, Zhenshan Bing, Carlo D'Eramo, Ozgur S Oguz, and Alois Knoll. Contact energy based hindsight experience prioritization. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5434–5440. IEEE, 2024.

[Schaul *et al.*, 2015] Tom Schaul, John Quan, Ioannis Antonoglou, et al. Prioritized experience replay. *arXiv preprint arXiv:1511.05952*, 2015.

[Schrittwieser *et al.*, 2020] Julian Schrittwieser, Ioannis Antonoglou, Thomas Hubert, Karen Simonyan, Laurent Sifre, Simon Schmitt, Arthur Guez, Edward Lockhart, Demis Hassabis, and Thore Graepel. Mastering atari, go, chess and shogi by planning with a learned model. *Nature*, 2020.

[Silver *et al.*, 2017] David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, et al. Mastering the game of go without human knowledge. *Nature*, 2017.

[Sujit *et al.*, 2023] Shivakanth Sujit, Somjit Nath, Pedro Braga, and Samira Ebrahimi Kahou. Prioritizing samples in reinforcement learning with reducible loss. *Advances in Neural Information Processing Systems*, 36:23237–23258, 2023.

[Sun *et al.*, 2020] Peiquan Sun, Wengang Zhou, and Houqiang Li. Attentive experience replay. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 5900–5907, 2020.

[Todorov *et al.*, 2012] 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*. IEEE, 2012.

[Wang *et al.*, 2024a] Yiming Wang, Ming Yang, Renzhi Dong, Binbin Sun, Furui Liu, et al. Efficient potential-based exploration in reinforcement learning using inverse dynamic bisimulation metric. *Advances in Neural Information Processing Systems*, 36, 2024.

[Wang *et al.*, 2024b] Yiming Wang, Kaiyan Zhao, Furui Liu, et al. Rethinking exploration in reinforcement learning with effective metric-based exploration bonus. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

[Wei *et al.*, 2021] Qing Wei, Hailan Ma, Chunlin Chen, and Daoyi Dong. Deep reinforcement learning with quantum-inspired experience replay. *IEEE Transactions on Cybernetics*, 52(9):9326–9338, 2021.

[Yang *et al.*, 2023a] Ming Yang, Renzhi Dong, Yiming Wang, Furui Liu, Yali Du, Mingliang Zhou, and Leong Hou U. Tiecomm: Learning a hierarchical communication topology based on tie theory. In *International Conference on Database Systems for Advanced Applications*, pages 604–613. Springer, 2023.

[Yang *et al.*, 2023b] Ming Yang, Yiming Wang, Yang Yu, Mingliang Zhou, et al. Mixlight: Mixed-agent cooperative reinforcement learning for traffic light control. *IEEE Transactions on Industrial Informatics*, 2023.

[Yang *et al.*, 2024] Ming Yang, Kaiyan Zhao, Yiming Wang, Renzhi Dong, Yali Du, Furui Liu, Mingliang Zhou, and Leong Hou U. Team-wise effective communication in multi-agent reinforcement learning. *Autonomous Agents and Multi-Agent Systems*, 38(2):36, 2024.

[Yarats *et al.*, 2021] Denis Yarats, Amy Zhang, Ilya Kostrikov, Brandon Amos, Joelle Pineau, and Rob Fergus. Improving sample efficiency in model-free reinforcement learning from images. In *Proceedings of the aaai conference on artificial intelligence*, volume 35, pages 10674–10681, 2021.

[Yu *et al.*, 2024] Jiayu Yu, Jingyao Li, Shuai Lü, and Shuai Han. Mixed experience sampling for off-policy reinforcement learning. *Expert Systems with Applications*, 251:124017, 2024.

[Zhang *et al.*, 2017] Jianan Zhang, Jost Tobias Springenberg, Joschka Boedecker, and Wolfram Burgard. Deep reinforcement learning with successor features for navigation across similar environments. In *Proceedings of the IEEE International Conference on Intelligent Robots and Systems (IROS)*, pages 5276–5281. IEEE, 2017.

[Zhao and Tresp, 2018] Rui Zhao and Volker Tresp. Energy-based hindsight experience prioritization. In *Conference on Robot Learning*. PMLR, 2018.

[Zhou *et al.*, 2019] Hongyao Zhou, Peng Zhang, Zhanxing Qiu, Haifeng He, and Wei Zhang. Multi-goal reinforcement learning: Learning to learn and exploring within a complex and sparse environment. *arXiv preprint arXiv:1902.06899*, 2019.