Artificial Intelligence and Human-Computer Interaction
Y. Ye and H. Zhou (Eds.)

doi:10.3233/FAIA250100

Uncertainty-Based Dynamic Weighted Experience Replay for Human-in-the-Loop Deep Reinforcement Learning

Xia TIAN[a], Yu KANG[a,1], Yunbo ZHAO[a,b,c], Yaqing ZHOU[a] and Pengfei LI[a]
[a] *Department of Automation, University of Science and Technology of China, Hefei, China*
[b] *Institute of Artificial Intelligence, Hefei Comprehensive National Science Center, Hefei, China*
[c] *Institute of Advanced Technology, University of Science and Technology of China, Hefei, China*
ORCiD ID: Yu KANG https://orcid.org/0000-0002-8706-3252

Abstract. Human-in-the-loop reinforcement learning (HIRL) enhances sampling efficiency in deep reinforcement learning by incorporating human expertise and experience into the training process. However, HIRL methods still heavily depend on expert guidance, which is a key factor limiting their further development and largescale application. In this paper, an uncertainty-based dynamic weighted experience replay approach (UDWER) is proposed to solve the above problem. Our approach enables the algorithm to detect decision uncertainty, triggering human intervention only when uncertainty exceeds a threshold. This reduces the need for continuous human supervision. Additionally, we design a dynamic experience replay mechanism that prioritizes machine self-exploration and human-guided samples with different weights based on decision uncertainty. We also provide a theoretical derivation and related discussion. Experiments in the Lunar Lander environment demonstrate improved sampling efficiency and reduced reliance on human guidance.

Keywords. Human-in-the-loop, human guidance, uncertainty, dynamic weighted

1. Introduction

Human-in-the-loop reinforcement learning (HIRL) is an approach that enhances the efficiency of deep reinforcement learning by incorporating human feedback during the learning process [1][2]. It has gained significant attention in recent years and has shown potential in areas like autonomous driving [3], robot control [4], and large language models [5], which has initially verified its effectiveness and prospects in complex tasks.

However, HIRL methods still heavily depend on expert guidance, which is a key factor limiting their further development and large-scale application [6]. The high cost of human guidance arises from the challenge of predicting unsafe machine actions, requiring constant supervision. Additionally, current methods often underutilize human guidance, further increasing reliance on human labor. For instance, HG-Dagger [7] lacks

[1] Corresponding Author: Yu KANG, kangduyu@ustc.edu.cn.

a weighting mechanism for human guidance data, and Li [8] fails to fully leverage human contributions by equally sampling human and proxy data.

Currently, research has begun to focus on reducing human guidance. One approach [9] is to train models to imitate human decisions, which works well in simple scenarios but struggles in complex ones. Other approaches use expert guidance with constrained rewards [3], but designing effective reward functions for real-world environments remains challenging. Thus, developing more efficient HIRL algorithms to reduce human guidance remains a key challenge.

Based on the above challenges, this paper introduces an innovative solution: the Uncertainty-based Dynamic Weighted Experience Replay (UDWER) approach. In the following sections, we will first explain the theoretical foundations and design of UDWER, and then demonstrate its effectiveness through experimental results. This solution addresses the limitations of current Human-in-the-loop methods by detecting uncertainty to engage experts only when needed and using a dynamic weighting mechanism, thus reducing human involvement and improving learning efficiency.

The rest of this paper is organized as follows: Section 2 presents the proposed method with theoretical validation, Section 3 discusses the experiments and results, and Section 4 concludes the study.

2. Methodology

2.1. Control transfer based on uncertainty

The DRL agent's control is modeled as a Markov decision process (MDP) (S, A, R, p, γ). Using the MC-dropout method [10], we measure the uncertainty of the machine's decision am, allowing the algorithm to output both the decision and its uncertainty [11]. Equation 1 describes how to calculate the uncertainty of the decision, where $\theta_i \sim p(\theta|s_{1:t-1}, a_{m1:t-1})$ represents the probabilistic network parameters.

$$E[a_m] \approx \frac{1}{M}\sum_{i=1}^{M} f^{\theta_i}(s) \tag{1a}$$

$$E[(a_m)^T a_m] \approx \frac{1}{M}\sum_{i=1}^{M}[\tau^{-1} I + f^{\theta i}(s)^T f^{\theta i}(s)] \tag{1b}$$

$$b = \mathrm{Var}[a_m] = E[a_m^T a_m] - E[a_m]^T E[a_m] \tag{1c}$$

In the interaction between DRL and the environment, the agent's strategy generates actions to explore the environment. Given a DDQN deep reinforcement learning agent, the strategy output is [12]:

$$a_t^{DRL} = \arg\max_a Q(s_i, a; \theta) \tag{2}$$

When the agent's decision uncertainty b_t (equation 1) exceeds a predefined threshold, a human expert intervenes to ensure safety. This mechanism—referred to as **control transfer**—allows the system to alternate between autonomous agent decisions and human interventions based on real-time uncertainty levels.

$$a_t = \begin{cases} \arg\max_a Q(s_i, a; \theta) & b_t < \lambda \\ a_t^H & b_t \geq \lambda \end{cases} \tag{3}$$

Among them, a_t^H represents the human expert's decision. If $b_t < \lambda$, the agent's uncertainty is within an acceptable range, and the agent makes the decision. If $b_t \geq \lambda$, uncertainty is too high, requiring human intervention. The resulting interaction tuple is stored in the experience replay buffer, where both human and machine actions are saved together, along with the agent's decision uncertainty for future use:

$$\mathcal{D} \leftarrow \zeta_i = (s_i, a_i, r_i, s_{i+1}, b_i) \tag{4}$$

2.2. Dynamic-weight experience replay

After establishing the mechanism for control transfer between agent and human decisions, we now focus on the learning process. We observe that machine self-exploration samples triggering human intervention are less effective for learning than human-guided samples. Thus, the learning weight of self exploration samples should be reduced, while human-guided samples should be prioritized. Although this difference is underappreciated in existing approaches, it provides new perspectives for more effective use of human-guided data.

Based on this, we propose a dynamic weighted experience replay mechanism, which adjusts the learning priority of different experiences based on their associated uncertainty. Human-guided experiences are given higher priority to enhance learning efficiency, while machine-generated experiences with high uncertainty are given lower priority.

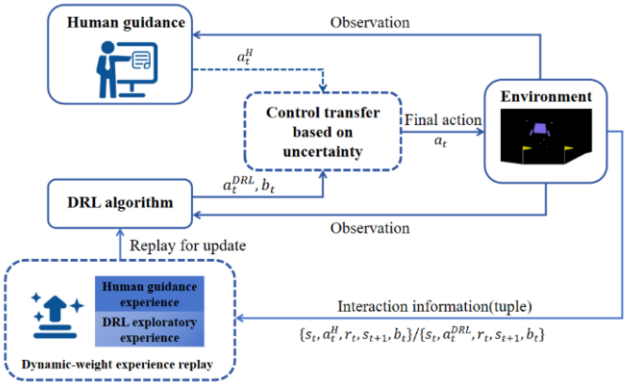


Figure 1. Uncertainty-based Dynamic weighted experience replay for Human-in-the-loop DRL.

A weighting function is created which maps the uncertainty level b_t of each experience tuple to a weight, meeting these conditions:

- Assign lower weights to machine decisions with high uncertainty.
- Increase weights for human-guided experiences to enhance learning.
- Normalize the final weights so their sum equals one, ensuring consistency.

In the experience buffer, let D_m be the set of experiences corresponding to machine -environment interactions, with n_m entries, and D_h be the set of experiences correspond ing to human-environment interactions, with n_h entries. The design weight function is:

$$\omega(b_i) = \begin{cases} \frac{\frac{b_i^{-1} n_m}{\sum_{i \in D_m} b_i^{-1}}}{n_m + n_h}, & b_i < \lambda \\ \frac{1}{n_m + n_h}, & b_i \geq \lambda \end{cases} \quad (5)$$

Proof. For the case $b_i < \lambda$, the sum of all these weights can be expressed as:

$$\sum_{i \in D_m} \frac{\frac{b_i^{-1} n_m}{\sum_{i \in D_m} b_i^{-1}}}{n_m + n_h} = \frac{\sum_{i \in D_m} b_i^{-1} n_m}{\sum_{i \in D_m} b_i^{-1} (n_m + n_h)}$$

Since the numerator denominator divides both $\sum_{i \in D_m} {b_i}^{-1}$, the sum of this part is 1. Dividing this result by $n_m + n_h$,we get $\frac{n_m}{n_m + n_h}$.For the case of $b_i \geq \lambda$, the weight of each experience is $(n_m + n_h)^{-1}$, so the sum of this part is:

Add the two parts together at the end:

$$\frac{n_m}{n_m + n_h} + \frac{n_h}{n_m + n_h} = \frac{n_m + n_h}{n_m + n_h} = 1$$

That is $\sum_{i \in (D_m \cup D_h)} \omega(b_i) = 1$, which satisfies the third condition.■

During learning, n_m, n_h represent the fixed ratio of human to agent decision data in the experience buffer. Carefully analysing the formula (5), for humans, it is usually assumed that the experience gained by individuals are not similar and can be considered as uniformly distributed, while the agent's learning weight $\omega(b_i)$ decreases as uncertainty b_i increases, reflecting the need for lower weights on uncertain agent decisions. This fulfils the first two conditions.

By choosing the appropriate parameter λ , machine decision weights are kept smaller than those from human interactions. These weights are dynamically adjusted based on real-time performance. The dynamic weighting function is then applied to the loss function to improve gradient descent and update the Q-network parameters.

$$L_{dw} = \omega(b_i) \cdot L_\theta = \omega(b_i) \cdot \frac{1}{N} \sum_{i=1}^{N} (r_i + \gamma \max_{a'} Q(s_{i+1}, a'; \theta^-) - Q(s_i, a_i; \theta))^2 \quad (6)$$

2.3. Workflow

As shown in Figure 1, the workflow of the method is as follows: The environment provides initial observation data to the DRL agent, which generates an action a_t^{DRL} and its uncertainty b_t. When $b_t \geq \lambda$, the human intervenes with an action a_t^H .The DRL agent and human then determine the final action a_t through an interactive control mechanism 2.1, which is applied to the environment. Interaction data, including state s_t ,action $a_t(a_t^{DRL}/a_t^H)$, reward r_t , next state s_{t+1} , and uncertainty b_t , are stored as tuples in the experience replay buffer. The agent updates its strategy using data from the experience replay.

3. Experiments

3.1. Experimental settings

This chapter uses OpenAI's Lunar Lander (Figure 2) as a simulation to verify the effectiveness of the proposed algorithm. The objective is to control the lunar lander to touch down in a designated area without crashing. The action space consists of four discrete options: no action, left engine, main engine, and right engine. Each episode lasts up to 1000 steps, with failures penalized by -100 points and successful landings rewarded with 100 points. A task is considered complete when the average reward reaches 200 over 100 trials.

The central assumption is that human operators make optimal decisions and act to maximize their benefits. During training, once human become familiar with the task, the agent explores and updates its strategy. We will evaluate learning performance improvements and associated labor costs by comparing three methods: UDWER, PHIL-TD3 [3], and DDQN. Metrics include cumulative returns, success rates, crash rates, and the human guidance needed.

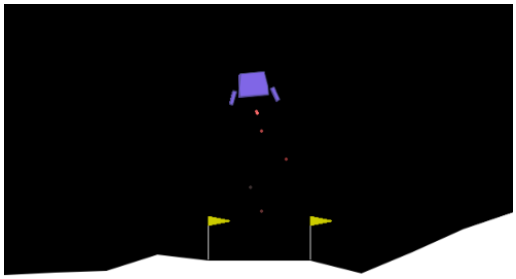


Figure 2. OpenAI Lunar Lander environment.

3.2. Discussions of experimental results

As shown in Figure 3, UDWER performed well throughout the training process, converging quickly and obtaining higher average returns for most of the time, while fluctuating less. PHIL-TD3 converged faster but had lower returns and more fluctuations. DDQN, in contrast, had the slowest convergence speed.

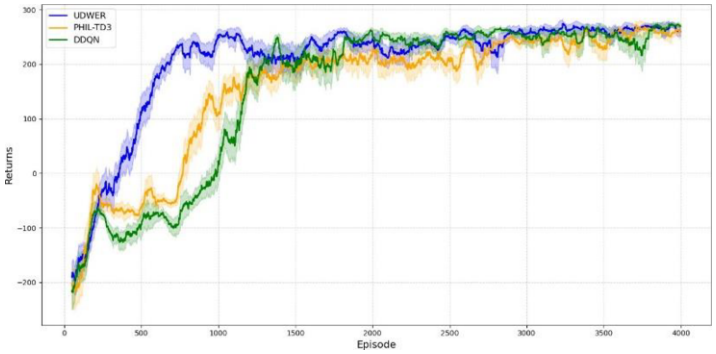
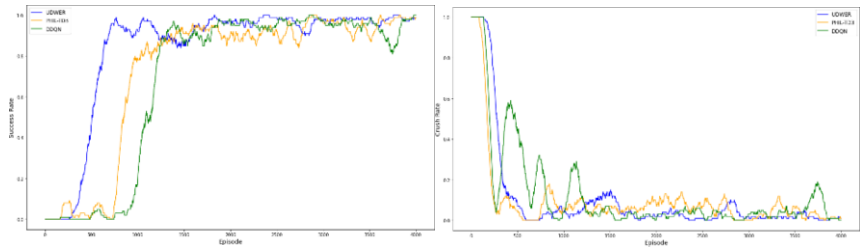


Figure 3. Return within 95% confidence interval.

A successful and safe landing is crucial for game success. As shown in Figure 4, UDWER quickly achieved a high success rate and low crash rate, stabilizing after 400 rounds (Figure 4(a)). PHIL-TD3 and DDQN showed a fluctuating rise or fall, which was slow and less stable.



(a)Success rate (b) Crash rate

Figure 4. Comparison of task completion.

Table 1 indicates UDWER had the highest win rate (84.85%), followed byPHIL-TD3 (74.33%) and DDQN (70.05%). While PHIL-TD3 had the lowest failure rate (8.30%), UDWER (9.00%) was more efficient overall, completing tasks with fewer timeouts. DDQN had the highest failure rate (11.45%). Overall, UDWER outperformed the others in task efficiency and success rate.

Table 1. Comparison of algorithm task completion.

	win	overtime	lose
UDWER	3394	246	360
PHIL-TD3	2973	695	332
DDQN	2802	740	458

Figure 5 compares human guidance costs between UDWER and PHIL-TD3 during the early stages of training. Both methods rely on human guidance to prevent unsafe actions, but differ in how they leverage it. During the 35-minute task, UDWER used human guidance in 32.9% of interactions, 8.2% lower than PHIL-TD3, showing more efficient use of guidance data. UDWER's efficiency comes from its uncertainty-based approach and dynamic weight experience replay, which reduces unnecessary interventions and integrates human input more effectively.

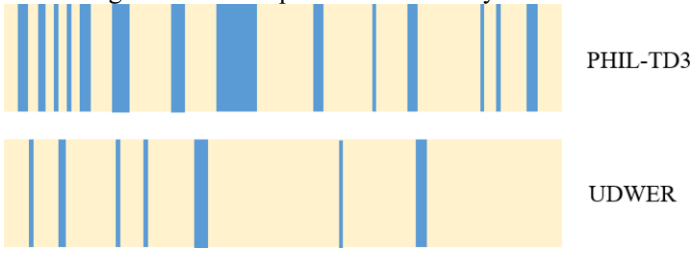


Figure 5. Comparison of human guidance costs in the early stages of training.

4. Conclusion

This paper introduces an Uncertainty-based Dynamically Weighted Experience Replay method tailored for Human-in-the-loop Deep Reinforcement Learning. Experimental results demonstrate that our approach effectively reduces the need for continuous human intervention while improving sampling efficiency. Our approach shows great potential for applications in areas such as autonomous driving and robotics.

Future research will focus on applying this approach to real-world scenarios and exploring its scalability in larger, more complex systems. Another key direction is the introduction of adaptive thresholds for human intervention, especially in environments with significant complexity variations, to improve decision-making efficiency.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (No. 62173317, No. 62103394), and the Zhuimeng Fund of Jianghuai Advanced Technology Innovation Center (No. Q/JH-063-2023-T02).

References

[1] Retzlaff CO, Das S, Wayllace C, et al. Human-in-the-Loop Reinforcement Learning: A Survey and Position on Requirements, Challenges, and Opportunities. Journal of Artificial Intelligence Research. 2024; 79:359–415, doi: 10.1613/jair.1.15348.

[2] Steffny L, Dahlem N, Reichl L, et al. Design of a Human-in-the-Loop Centered AI-Based Clinical Decision Support System for Professional Care Planning. HHAI 2023: Augmenting Human Intellect [Internet]. IOS Press; 2023 [cited 2024 Oct 2]. p. 263–273, doi: 10.3233/FAIA230090.

[3] Wu J, Huang Z, Huang W, et al. Prioritized Experience-Based Reinforcement Learning with Human Guidance for Autonomous Driving. IEEE Transactions on Neural Networks and Learning Systems. 2024;35(1):855–869, doi: 10.1109/TNNLS.2022.3177685.

[4] Andersson SKL, Granlund A, Hedelind M, et al. Exploring the Capabilities of Industrial Collaborative Robot Applications. SPS2020 [Internet]. IOS Press; 2020 [cited 2024 Oct 2]. p. 109–118, doi: 10.3233/ATDE200148.

[5] Ouyang L, Wu J, Jiang X, et al. Training language models to follow instructions with human feedback. Advances in neural information processing systems. 2022; 35:27730–27744, doi: 10.48550/arXiv.2203.02155.

[6] MacGlashan J, Ho MK, Loftin R, et al. Interactive learning from policy-dependent human feedback. International conference on machine learning [Internet]. PMLR; 2017 [cited 2024 Oct 2]. p. 2285–2294, doi: 10.48550/arXiv.1701.06049.

[7] Kelly M, Sidrane C, Driggs-Campbell K, et al. HG-DAgger: Interactive Imitation Learning with Human Experts. 2019 ICRA). p. 8077–8083, doi: 10.1109/ICRA.2019.8793698.

[8] Li Q, Peng Z, Zhou B. Efficient Learning of Safe Driving Policy via Human-AI Copilot Optimization [Internet]. arXiv; 2022, doi: 10.48550/arXiv.2202.10341.

[9] Hilleli B, El-Yaniv R. Toward Deep Reinforcement Learning Without a Simulator: An Autonomous Steering Example. Proceedings of the AAAI Conference on Artificial Intelligence [Internet]. 2018 [cited 2024 Oct 2];32(1), doi: 10.1609/aaai.v32i1.11490.

[10] Gal Y, Ghahramani Z. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. international conference on machine learning [Internet]. PMLR; 2016. p. 1050–1059, doi: 10.48550/arXiv.1506.02142.

[11] Zhang Q, Kang Y, Zhao Y-B, et al. Traded Control of Human–Machine Systems for Sequential Decision-Making Based on Reinforcement Learning. IEEE Transactions on Artificial Intelligence. 2022;3(4):553–566, doi: 10.1109/TAI.2021.3127857.

[12] Van Hasselt H, Guez A, Silver D. Deep reinforcement learning with double q-learning. Proceedings of the AAAI conference on artificial intelligence [Internet]. 2016;30(1), doi: 10.1609/aaai.v30i1.10295.