TABLE 1 Hindsight experience replay mechanisms.

Algorithm 1 Hindsight experience replay

Initialise $\mathbb{A}$

Initialise replay buffer $\boldsymbol{D}$

for episode = 1, M **do**

Sample a goal g and an initial state s_0.

for $t = 0$, $T-1$ **do**

Sample an action a using the behavioural policy from: $\mathbb{A}$

$a_t \leftarrow \pi_b(s_t \parallel g)$

Execute the action a and observe a new state s'

end for

for t = 0, T−1 **do**

$r = r(s, a, o, g)$

Store the transition $(s \parallel g, a, o \parallel g, r, s' \parallel g)$ in $\boldsymbol{D}$

Sample a set of additional goals for replay $G := \mathbb{S}$(currentepisode)

for$g' \in G$ **do**

$r' := r(s, a, o, g')$

Store the transition $(s \parallel g', a, o \parallel g', r', s' \parallel g')$ in $\boldsymbol{D}$

end for

end for

for $t = 1$, N **do**

Sample a minibatch B from the replay buffer $\boldsymbol{D}$

Perform one step of optimisation using $\mathbb{A}$ a minibatch B

end for

end for

TABLE 2 LSTM-DQN-HER learning algorithm based on the two-player zero-sum game in three-dimensional space.

Algorithm 2 LSTM-DQN-HER

1. Initialise the HER experience playback pool $\boldsymbol{D}$ with a capacity of N
2. Create two neural networks: LSTM Q network and target LSTM network. The parameters of the LSTM Q network are θ as follows:

The target network parameters are; $\theta^- = \theta$

3. **for** episode = 1,2,…, M **do**:
4. Initialisation: state of the environment; S_1
5. **for** step = 1,2,…,T **do**.
6. Generate actions using the $\varepsilon - greedy$ strategy a
7. Execute the action a, get the next state s, receive the reward r, and observe the strategy o executed by Blue;
8. Update $Q(s, a, o)$: $Q(s, a, o) = R(s, a, o) + \gamma \sum_{s'} T(s, a, o, s')V(s')$
9. Solve using linear programming: $V(s) = \max_{\pi \in \mathrm{PD}(A)} \min_{o \in O} \sum_{a \in A} Q(s, a, o)\pi_a$ using argmax and update $V(s)$ and $\pi_a(s)$;
10. Sample transitions from the moment i$(s \parallel g, a, o \parallel g, r, s' \parallel g)$ are stored in the HER experience playback pool $\boldsymbol{D}$;
11. Randomly select a sample of minibatch transitions from the HER experience playback pool $\boldsymbol{D}$. $(s \parallel g', a, o \parallel g', r', s' \parallel g')$
12. If the step $j + 1$ is the final moment, then $y_j = r_j$

If the $j + 1$ step is not the final moment, then the $y_j = r_j + \gamma Q(s', a', \theta^-)$

13. Update $\mathrm{loss}(\theta) = \frac{1}{2}[r(s) + \gamma Minmax Q(s', a', o', \theta^-) - Q(s', a', o', \theta)]^2$ using gradient descent;
14. LSTM units are implemented for network learning using Equation (21) and F.C. layer network learning using Equations (22), (23) and (24);

Update the network node status;

15. Update the target network every C step $\theta^- = \theta$,
16. **End for**
17. **End for**

of a fighter jet. To solve the relevant posture information, we employed the Eulerian method to handle the dynamic function with a 100 Hz action stepping period, the state is observed by the agent every 10 Hz. The algorithm proposed in this paper is a value function learning method based on discrete action space. Agent 1's decision-making rate for manoeuvres is limited to 1 Hz and it employs the TZSG LSTM-DQN-HER algorithm. On the other hand, Agent 2 executes manoeuvre actions based on a manoeuvre strategy state machine. As a control experiment, agent 2's strategy remains constant while agent 1 uses DQN, DDQN, Dueling DQN, and LSTM-DQN algorithms to select actions. The hyperparameters for each algorithm are set as shown in the table below, shown in Table 3:

The construction of the three-dimensional TZSG posture is illustrated in Figure 9. Figure 9a displays the initial unfavourable posture of the algorithm, which validates its adaptability. Figure 9b portrays the initial advantageous posture of the algorithm, which confirms its effectiveness, Tables 4 and 5 show the initial parameters of the agent.

The spatial coordinates of agent are $\boldsymbol{c}_i (i \in R, B)$, and the inclination and declination of agent's trajectory are $\gamma_i (i \in R, B)$, $\psi_i (i \in R, B)$, $v_i (i \in R, B)$ are velocity size, the initial health value, both of them which is 3. The end game condition of each round is that either side's health value goes to 0, measured by the most extended duration of a game if no one wins. The initial health value is 3. The endgame condition for each round is that either player's health goes to 0, and the most extended game duration is measured when no one wins. This is usually set to 300 s.

5.2 | Training results

We incorporated the LSTM-DQN-HER method with the five core algorithms of deep reinforcement learning, which rely on value functions as discussed in Section 5.1. These algorithms were utilized for training purposes in the 3D TZSG environment. We successfully obtained training results in both disadvantageous and advantageous states, as illustrated in Figure 10a,b.

By analysing the experimental results, we present here two main findings.

To enhance the convergence speed of a deep reinforcement learning algorithm based on the value function for solving the 3D TZSG problem, the addition of an LSTM network

TABLE 3 Hyperparameters of each algorithm.

	α	γ	$\varepsilon - greedy$	tau	Timestep	Batch size	Capacity
LSTM-DQN-HER	0.001	0.9	0.99	0.005	100	512	1,000,000.
LSTM-DQN	0.001	0.9	0.99	0.005	100	512	1,000,000.
DDQN	0.002	0.95	0.99	0.005	None	1024	1,000,000.
DQN	0.001	0.91	0.99	None	None	2048	1,000,000.
Dueling DQN	0.003	0.93	0.99	0.005	None	1024	1,000,000.

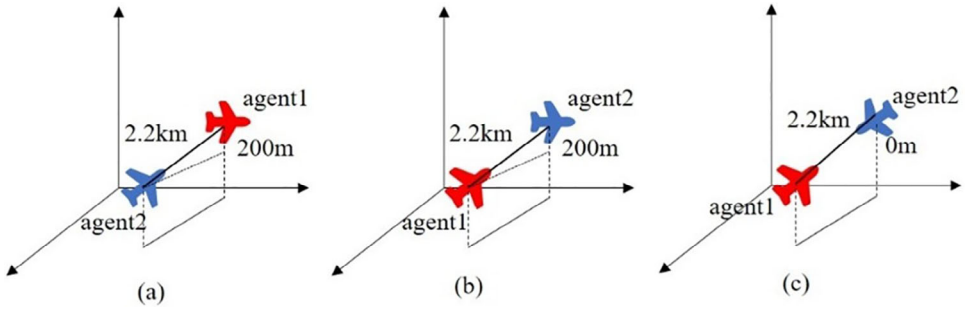

FIGURE 9 Initial posture.

TABLE 4 Agent1 initial settings.

	c_R	γ_R	ψ_R	v_R	Blood
Figure 9a	(−1997.48, 0.04, −6000.19)	0°	0°	350 m/s	3
Figure 9b	(−0.04, 2.50, −7000.19)	0°	270°	350 m/s	3
Figure 9c	(−0.04, 2.50, −7000.19)	0°	0°	350 m/s	3

TABLE 5 Agent2 initial settings.

	c_B	γ_B	ψ_B	v_B	Blood
Figure 9a	(−0.04, 2.50, −7000.19)	0°	270°	350 m/s	3
Figure 9b	(−1997.48, 0.04, −6000.19)	0°	0°	350 m/s	3
Figure 9c	(−1997.48, 0.04, −6000.19)	0°	180°	350 m/s	3

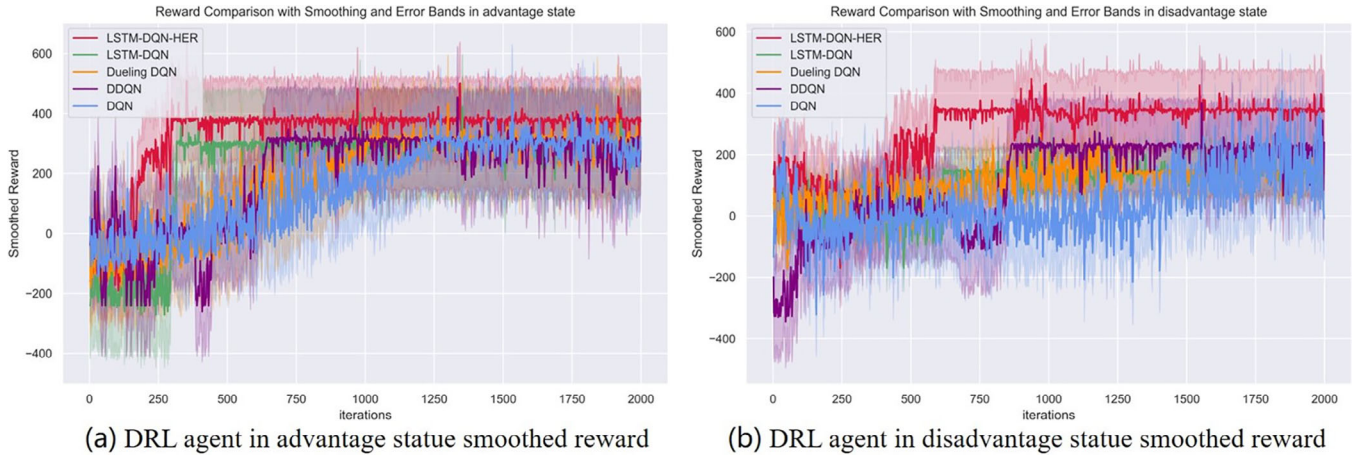

FIGURE 10 Average normalised episodic rewards for the adversary training experiment using state machines. Using one machine (Intel Core i9-13900K CPU (5.0GHz, 24 cores), 1 NVIDIA GeForce RTX 4080 16G GPU, 64GB RAM).

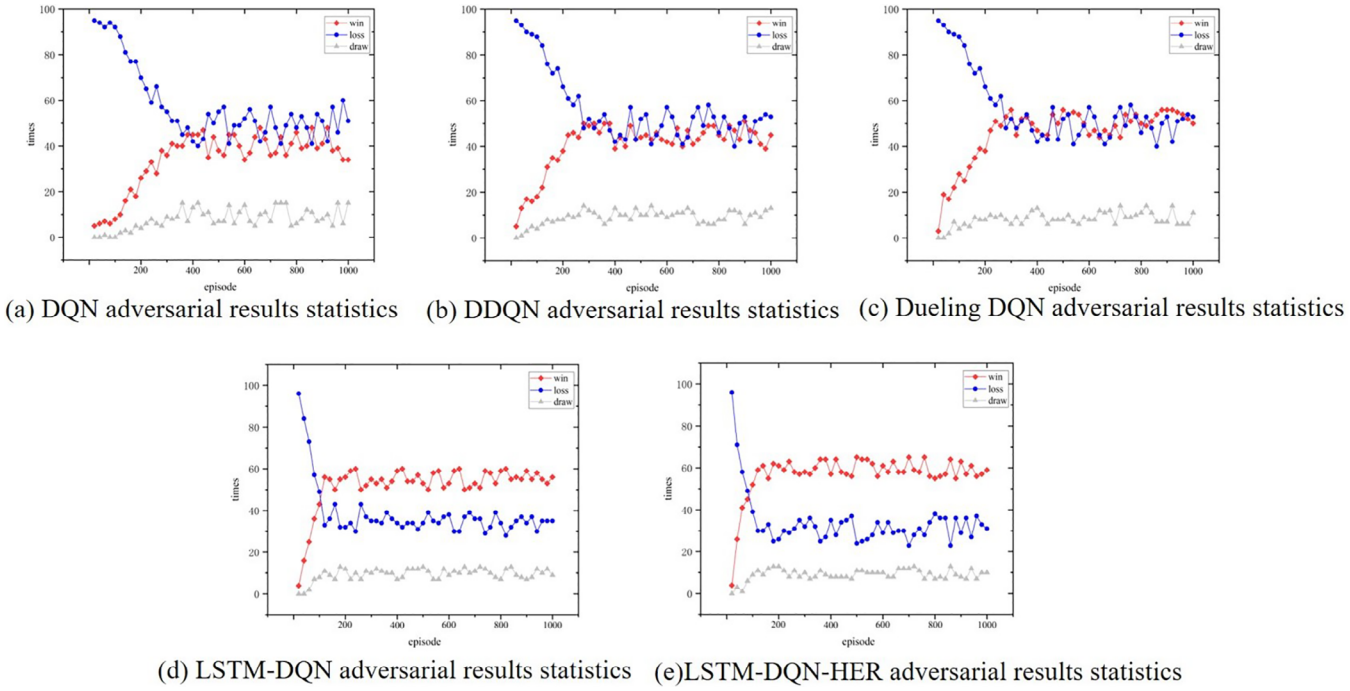

FIGURE 11 Confrontation curve of DRL algorithm with gaming expert system.

(shown in Figure 10a,b) effectively improves convergence. Both algorithms that incorporate the LSTM achieve early convergence, with the LSTM-DQN-HER converging at the 250th iteration and the LSTM-DQN converging at the 300th iteration in the advantageous case. In the disadvantageous case, the LSTM-DQN-HER converges at the 600th iteration, and the LSTM-DQN converges at the 700th iteration. This demonstrates that the 3D TZSG problem is influenced by the timing of actions and states.

When solving the 3D TZSG problem using the same DRL algorithm, the introduction of the empirical processing mechanism of HER (as shown in Figure 10a,b) leads to better algorithmic performance metrics. Specifically, the rewards obtained by LSTM-DQN-HER are higher than those of the LSTM-DQN algorithm, regardless of the game situation. Furthermore, the LSTM-DQN-HER algorithm, with the HER mechanism in place, is the only one capable of achieving the final algorithmic metrics in the dominant state, even when starting from an inferior initial state.

5.3 | Adversarial game experiments with expert systems

In addition to algorithm convergence, evaluating the effectiveness of the algorithm in the two-player zero-sum game problem also takes into account the winners and losers of the game. In the experiments conducted in this paper, the neural network saves the model every 1000 iterations of learning. After the training is completed, the model is loaded sequentially according to the chronological order shown in Figure 9c. Different expert system opponents are then used to play the game for 100 rounds, and the win/loss outcomes of each confrontation are statistically calculated. The experimental results are presented in Figure 11a–e.

In Figure 11a, the DQN algorithm demonstrates an average win rate of approximately 40%. The wins do not exhibit significant improvement beyond the 400th episode, indicating convergence of the algorithm. Moving on to Figure 11b, the DDQN algorithm achieves an average win rate of around 43% and converges at approximately the 300th episode. Similarly, the Dueling DQN algorithm in Figure 11c attains an average win rate of about 45% and converges at around the 300th episode. In Figure 11d, the LSTM-DQN algorithm showcases an average win rate of about 52% and converges at about the 180th episode. Lastly, in Figure 11e, the LSTM-DQN-HER algorithm achieves an average win rate of approximately 60% and converges at around the 150th episode. These results demonstrate that the LSTM-DQN-HER algorithm exhibits strong performance and robustness in solving the 3D TZSG problem.

5.4 | Behavioural emergent analysis

In the context of deep reinforcement learning, it is important to not only analyse the quantitative data but also incorporate visual analysis to gain a more comprehensive understanding of the training results. By designing the training environment appropriately, we can visually assess the outcomes. During our observations and analysis of various test simulations, we have identified a common dogfighting pattern that occurs when the DRL Agent holds a dominant position. This pattern involves

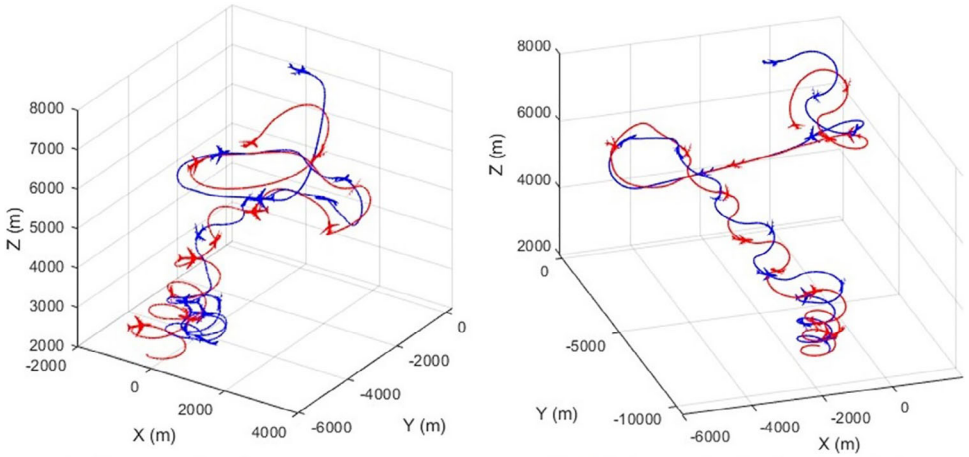

FIGURE 12 Manoeuvring trajectories of the R.L. Agent game at different initial conditions.

the effective use of a "vertical manoeuvre" tactic by the DRL Agent, as shown in Figure 12a. In this particular scenario, both sides engage in an intense struggle and pursuit, leading to a decrease in altitude. Subsequently, a risky "low-altitude dogfighting process" takes place, where the RL Agent aims to maintain a position directly behind the opponent, thereby reducing the opponent's health or causing mistakes such as poorly controlled altitude that may result in near-ground collisions or forced pull-ups.

In our review and analysis, we found that when at a disadvantage, the DRL Agent learns a manoeuvre strategy (shown in Figure 12b) that allows it to turn its disadvantageous position into a balanced one. This strategy involves a process called "vertical manoeuvre," where the DRL Agent ascends vertically, performs a rolling inversion manoeuvre, and then quickly follows it with a continuous turning manoeuvre to outmanoeuvre the adversary and gain an advantage in the process.

6 | CONCLUSIONS

In this study, we have proposed an innovative deep reinforcement learning solution, named LSTM-DQN-HER, for a two-player zero-sum game conducted in three-dimensional space. Our solution enhances the existing LSTM-DQN algorithm by integrating the LSTM layer of recurrent neural networks and the HER mechanism. Through our training experiments, we demonstrate the effectiveness of the LSTM-DQN algorithm enhanced with LSTM and HER. In comparison to other deep reinforcement learning methods that rely on value functions, our approach significantly improves learning efficiency and algorithm performance. Moreover, our test results reveal that the strategies learned using the LSTM-DQN-HER algorithm exhibit strong emergence and generalization capabilities. These strategies empower the agent to consistently and swiftly navigate through combinatorial actions to achieve a favourable position. This study not only provides a viable approach for solving two-player zero-sum game problems in a three-dimensional space using deep reinforcement learning based on value functions but also offers valuable insights for enhancing the performance of DRL algorithms.

AUTHOR CONTRIBUTIONS

Bo Lu: Conceptualization; formal analysis; funding acquisition; investigation; methodology; resources; software; visualization; writing—original draft; writing—review and editing. **Le Ru**: Conceptualization; funding acquisition; methodology; project administration; resources; software; supervision; writing—review and editing. **Maolong Lv**: Data curation; methodology; supervision; writing—review and editing. **Shiguang Hu**: Software; supervision; validation; writing—review and editing. **Hongguo Zhang**: Resources; software; validation. **Zilong Zhao**: Supervision; writing—review and editing.

ACKNOWLEDGEMENTS

The authors would like to express our sincere gratitude to all the individuals who have supported us in conducting this work, as well as the reviewing committee for their valuable feedback.

CONFLICT OF INTEREST STATEMENT

The authors declare no conflicts of interest.

DATA AVAILABILITY STATEMENT

The data that support the findings of this study are available from the corresponding author upon reasonable request.

ORCID

Bo Lu https://orcid.org/0009-0007-8285-776X
Shiguang Hu https://orcid.org/0000-0001-6819-4700

REFERENCES

1. Virtanen, K., et al.: Modeling air combat by a moving horizon influence diagram game. J. Guid. Control Dyn. 29(5), 1080–1091 (2006). https://doi.org/10.2514/1.17168