IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 20 August 2022, accepted 23 October 2022, date of publication 26 October 2022, date of current version 4 November 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3217654

RESEARCH ARTICLE

Deep Reinforcement Learning Task Assignment Based on Domain Knowledge

JIAYI LIU, GANG WANG, XIANGKE GUO, SIYUAN WANG, AND QIANG FU
Air Defense and Antimissile School, Air Force Engineering University, Xi'an, Shaanxi 710051, China
Corresponding author: Qiang Fu (fuqiang_66688@163.com)

This work was supported by the Project of National Natural Science Foundation of China under Grant 62106283 and Grant 72001214.

ABSTRACT Deep Reinforcement Learning (DRL) methods are inefficient in the initial strategy exploration process due to the huge state space and action space in large-scale complex scenarios. This is becoming one of the bottlenecks in their application to large-scale game adversarial scenarios. This paper proposes a Safe reinforcement learning combined with Imitation learning for Task Assignment (SITA) method for a representative red-blue game confrontation scenario. Aiming at the problem of difficult sampling of Imitation Learning (IL), this paper combines human knowledge with adversarial rules to build a knowledge rule base; We propose the Imitation Learning with the Decoupled Network (ILDN) pre-training method to solve the problem of excessive initial invalid exploration; In order to reduce invalid exploration and improve the stability in the later stages of training, we incorporate Safe Reinforcement Learning (Safe RL) method after pre-training. Finally, we verified in the digital battlefield that the SITA method has higher training efficiency and strong generalization ability in large-scale complex scenarios.

INDEX TERMS Deep reinforcement learning, imitation learning, knowledge rule base, safe reinforcement learning, task assignment.

I. INTRODUCTION

Gaming refers to the process by which one or more rational players, under specific rules, choose and execute their respective sets of strategies to achieve corresponding gains.Large-scale game confrontation is a continuous decision-making process that requires better adaptive decisions in response to changes in the sparring situation. Task assignment is one of the key issues. Its primary purpose is to assign each task to the appropriate elements to perform to achieve the interception of the target and maximize the efficiency ratio of resources, which is a typical sequential decision-making process for non-complete information games [1]. Deep Reinforcement Learning (DRL) is a combination of Deep Learning (DL) and Reinforcement Learning (RL), which provides a new and efficient method for solving non-complete information gaming problems. It turns training into a data-driven self-supervised learning problem with good results in real-time strategy games and autonomous driving [2], [3]. However, many challenges remain in applying DRL to the task assignment of Large-scale game confrontation. For example, the decision-making process will be faced with a high-dimensional state-action space due to the complexity and variability of the combat situation and the number of entities involved. This significantly reduces the efficiency of interactive trial-and-error mechanisms for RL, even in complex task environments where effective strategies cannot be learned. Task goals are challenging to translate directly into proper reward functions that provide immediate and accurate feedback, resulting in behaviors facing sparse, late-and-inaccurate feedback.

In the above case, a more direct way is to use the decision data of many human experts to learn and thus obtain the agent's strategies, and such a method is called Imitation Learning (IL). IL investigates how to learn from expert decision examples to get decision models close to the expert level. With sufficient demonstration data, IL can quickly learn a strategy similar to the demonstration, which is a suitable learning method [4]. However, there are still many challenges to applying IL to large-scale game confrontation scenarios. For example, the need for a demonstration strategy or human

The associate editor coordinating the review of this manuscript and approving it for publication was Muhammad Afzal.

manual marking of each state throughout the training, which is difficult to achieve in a large-scale game confrontation scenario. The quality of the demonstration sample also severely limits the quality of learning, and the physical environment must be better mapped to the virtual environment. After obtaining the demonstrative sample, the long sequence of samples must be learned efficiently. After pre-training the neural network by IL, effective exploration must also be performed to continue improving the decision making of the agent.

This paper aims to apply the SITA method to task assignment in large-scale complex scenarios to improve the training efficiency of DRL in such scenarios. The problem of too much ineffective exploration in the early stages of DRL is first addressed through ILDN. Combined with the Safe RL approach, the demonstration policy is optimised to allow the agent to reach the desired decision-making level. To address the problem that it is challenging to obtain demonstration samples for large-scale game confrontation scenarios, this paper constructs a Knowledge and Rule Base (KRB) based on red-blue games, which transforms human knowledge and confrontation rules into a knowledge rule base and is used to replace the neural network in the agent. In order to ensure the quality of the demonstration sample, we built a high-simulation gaming adversarial environment for IL sampling and Safe RL training. In order to improve the learning efficiency of long sequence samples, the ILDN method is proposed in this paper and combined with the Safe RL method after pre-training to further stabilize and improve the decision making of the agent;Finally, the feasibility and superiority of the SITA method for large-scale complex scenarios are experimentally verified in the digital battlefield by taking the large-scale red-blue game task assignment problem as an example.

II. RELATED WORK

A. DEEP REINFORCEMENT LEARNING

The idea of RL is to use trial-and-error methods and rewards to train agents to learn behavior, and its essential environment is a Markov Decision Process(MDP). An MDP contains five quantities, namely $\langle S, A, r, P, \gamma \rangle$. Where S is a finite set of states, A is a limited set of actions, r is the reward function, P is the state transfer probability, and γ is the discount factor. The agent senses the current state of the environment and then makes the corresponding action to get the corresponding reward. However, traditional RL algorithms' inherent storage complexity, computational complexity, and sampling complexity make them suffer from dimensional catastrophe in large-scale complex environments. The combination of RL and DL, using deep neural networks as function fitters, gave birth to DRL [5], which effectively solved the problem of dimensional catastrophe [6]. In recent years, DRL has achieved good results in several fields, such as real-time strategy games [7], autonomous driving [8], and network resource optimization [9]. However, it is difficult to be applied in large-scale game confrontation, and the main challenges are insufficient data and the high cost of experimental validation. Therefore, this paper aims to create a high-simulation confrontation environment, which maps the physical environment to the virtual environment better and provides the foundation for the agent's training.

B. IMITATION LEARNING

IL studies how to learn from expert decision examples to obtain decision models close to the expert level. IL can get more direct feedback from decision examples and can be divided into Behavioral Cloning (BC) [10] and Inverse Reinforcement Learning (IRL) [11]. IRL first determines the reward function based on the given sample and then obtains the optimal policy, which indirectly reduces the expert knowledge and has strong generalization and robustness [12]. However, IRL tends to consume a large number of computational resources when solving large-scale complex problems, so it is not suitable for the scenario in this paper.

The main idea of BC is to directly clone the single-step action mapping of an expert sample at each state, i.e., to perform supervised learning on the expert sample. The prerequisite for BC to perform well is the availability of sufficient samples. The difficulty of sampling is precisely one of the main challenges in applying IL to the field of large-scale game confrontation. Therefore, this paper addresses this problem by transforming domain knowledge and sparring rules into a knowledge rule base to interact with the environment and provide the required samples for IL methods.

C. SAFE REINFORCEMENT LEARNING

Safety is a hot research topic in RL, and many scholars have conducted many studies on it. On top of RL, the goal of safe RL is to find a strategy that maximizes the expected value of the cumulative rewards of the agent on top of satisfying a predetermined set of safety constraints. According to constraint function $C_i : S \times A \rightarrow R, i = 1, 2, \ldots, m$, the agent receives a reward signal and a constraint signal. The commonly used solution methods are trust region policy optimization algorithm [13], constrained cross-entropy method [14], policy search method based on metric policy variability [15], constrained policy optimization (CPO) algorithm [16] and Lyapunov method [17], etc. The large-scale game confrontation scenario has many constraints and strict requirements for action safety, so this paper uses safe RL to ensure the safety of RL after the agent pre-training effectively.

D. TASK ASSIGNMENT

Task assignment in game confrontation is achieving optimal resource utilization by subdividing the various aspects of the conflict into multiple tasks and rationalizing the assignment among the different units (e.g., sensors and interceptors). The commonly used methods for solving task assignment are mainly intelligent optimization algorithms such as genetic algorithms [18] and simulated annealing

algorithms [19]; swarm intelligence algorithms such as ant colony algorithms [20] and fish swarm algorithms [21], and market mechanism-based methods such as auction algorithms [22] and contract network protocols [23].

With the increasing diversity of sparring situations, methods for generating deterministic strategies gradually fail to meet the demand. As DRL has both faster reactivity and higher adaptability, some scholars have used it to solve task assignment problems for large-scale complex scenarios. Still, the high-dimensional state-action space in it reduces the efficiency of DRL interactive trial-and-error, resulting in behaviors facing feedback sparsity, delays, and inaccuracies [24]. This paper proposes the SITA method, which first uses the ILDN method for pre-training to solve the cold start problem of DRL so that it has a better initial policy, and then combines with the SRL-GC method to continuously optimize the policy with constraints to solve the large-scale task assignment problem in complex scenarios.

III. PROBLEM MODELING

A. INTERACTION ENVIRONMENT

The agent must interact with the environment to obtain reward values during training, so the physical environment must be better mapped to the virtual environment. We modeled the training environment in a targeted manner and built a highly simulated digital battlefield.

The environmental interaction in this problem is in the form of a confrontation between the red and blue sides. The red side is the defender, responsible for defending strongholds and airfields, and its control units are mainly sensors and interceptors. The sensor contains two behaviors specific to itself, tracking the target and providing guidance information to the missile. It is also responsible for assigning individual targets to the missiles when simultaneously launching multiple missile attacks against multiple targets. The sensor also needs to confirm that the interception of the target was successful. The unique behavior of the interceptor is to ensure the surrounding sensors and use the proximity and long-range rounds appropriately for the distance to the target; once the missile is launched, the interceptor hands over control of the missile to the sensors.

The blue side is the attacking side, responsible for destroying strongholds, and the central control units are all various flying machines. The generic behavior of these aircraft in a variety of attitude adjustments, including the usual flight maneuvers such as cruise, climb, and dive, as well as ultra-low altitude assaults, turns towards the target, formation of attack conditions, missile launch, and evasive maneuvers to disengage from the target, and other special tricks. Other actions of the aircraft include using sensors to spot targets, searching for Red sensors, firing anti-radiation missiles to attack sensors, and firing air-to-ground missiles to attack Red protected objects while also being responsible for confirming that the missiles hit Red units (which may be done by other aircraft).

B. PROBLEM FORMULATION

This paper investigates the task assignment problem for the defender in a red-blue game confrontation scenario to use the least amount of resources when the protected object is least damaged. This paper solves the problem based on DRL to find an optimal policy π^* to maximize the expected value of the cumulative reward of an agent in an infinite time domain.

$$\pi^* = \arg\max_{\pi} E \sum_{t=0}^{\infty} \gamma^t r_t \left(s_t, \pi\left(s_t\right)\right) \tag{1}$$

where π is the set of policies, s_t is the state at moment t, r_t is the reward value at moment t, and γ^{t} is the discount factor at moment t.

C. MDP MODELING

To satisfy the rationality and completeness of the state space and action space and meet the needs of the game confrontation scenario, the state space, action space, and reward function in this paper are designed with reference to the literature [25].

State space: 1) state information of red protected objects; 2) state information of red interception units, including resource configuration, sensor and interceptor state, state information of blue targets within the interception range of the unit; 3) state information of blue units; 4) state information of blue units that can be attacked.

Action space: 1) which sensor to choose; 2) which interceptor to choose; 3) which blue targets to choose; 4) what timing to choose to intercept.

The reward function:

$$r = \begin{cases} 5m + n\text{-}0.05i & \text{Fail} \\ 50\text{+}5m + n - 0.05i & \text{Win} \end{cases} \tag{2}$$

In Equation (2), m is the number of intercepted blue manned units, n is the number of intercepted blue UAVs, and i is the number of missiles fired. Add 5 points for intercepting manned targets such as blue fighters, 1 point for intercepting UAVs, and the rest will not be treated as points; deduct 0.05 points for each missile fired, and add 50 points for obtaining the final victory. During DRL training, the reward value for the current moment r_t is fed back into Equation (1) according to Equation (2).

IV. METHOD

A. ACQUISITION OF DEMONSTRATION DATA

1) KNOWLEDGE AND RULE BASE (KRB) FOR RED AND BLUE GAME CONFRONTATION

Large-scale ground-to-air confrontation task assignment requires handling many concurrent tasks and random events, and the entire battlefield situation is full of complexity and uncertainty. The task assignment scheme in this paper is mainly based on the target threat estimation, with the criteria of eliminating the target with the highest threat value or the highest value, the highest probability of killing the target, and the lowest consumption of resources [26]. Priorities are

established based on expert knowledge and scenario elements to construct KRB, and some rules are as follows:

(1) The target is divided into levels according to the threat level: the threat level is divided into 0-10 levels according to the time when the target reaches the protected object and increases by one level for every 15s reduction.
(2) The highest priority is intercepting targets with high threat levels.
(3) When the anti-radiation missile and surface-to-air missile threat level reaches 6, one anti-aircraft missile is sent to intercept and enter the observation phase; if not killed, two more missiles are sent to intercept when the threat level reaches 10.
(4) When the threat level of the drone reaches 7 or more, send a missile to intercept, enter the observation phase, if not killed, when the interception conditions are met, send a missile to intercept again.
(5) When the fighter threat level reaches 7 or more, send one missile to intercept, enter the observation phase, and if no-kill is made, send two missiles to intercept again when the interception conditions are met.
(6) When the bomber threat level reaches 4 or more, send one missile to intercept, and if the threat level reaches 9, send two missiles to intercept.
(7) For anti-radiation missiles, priority is given to self-defense strategy interception, i.e., an interception by the interceptor unit under attack. When that unit cannot perform the interception, the other nearest interceptor unit assists in the interception.
(8) For Blue's cruise missiles, priority is given to intercepting them using close-range munitions.
(9) When intercepting a target, priority is given to the interceptor unit with a high probability of killing that target.
(10) When the probability of killing is the same, the unit whose sensors are tracking the target is used first to intercept.
(11) If more than one unit is tracking the target and has the same kill probability, the interceptor unit with the most ammunition remaining is assigned first.

2) STRUCTURE OF THE DEMONSTRATION DATA

Let the initial battlefield posture vector be $S_0 = (s_1^{(0)}, s_2^{(0)}, \cdots, s_{m1}^{(0)})$, and the battlefield posture vector in the k^{th} stage be the m_k dimensional vector $S_k = (s_1^{(k)}, s_2^{(k)}, \cdots, s_{mk}^{(k)})$. In the k^{th} phase Red has h_k optional actions $\left\{ra_1^{(k)}, ra_2^{(k)}, \cdots, ra_{\mathrm{h}_k}^{(k)}\right\}$ and Blue has l_k optional actions $\left\{ba_1^{(k)}, ba_2^{(k)}, \cdots, ba_{\mathrm{l}_k}^{(k)}\right\}$, the action strategy of the red side in the k^{th} phase is to choose n of the h_k optional actions to execute, defined as the vector $ra^{(k)}$, the components in the vector are {0,1} variables, where 0 means do not execute the action and 1 means execute the action. The optional action strategies that satisfy the resource constraint on the red side are $u_i^{(k)}$, and all optional action strategies constitute the strategy set $U^{(k)} = \left\{u_1^{(k)}, u_2^{(k)}, \cdots, u_{\mathrm{f}_k}^{(k)}\right\}$ in the k^{th} phase, and similarly define the strategy set $V^{(k)} = \left\{v_1^{(k)}, v_2^{(k)}, \cdots, v_{\mathrm{f}_k}^{(k)}\right\}$ on the blue side.

The battlefield situation in the k^{th} phase is the result of the game in the previous phase and the game condition in the current phase. The game of both sides in the k^{th} phase can be represented as a matrix response $G^{(k)} = \left\{U^{(k)}, V^{(k)}, \cdots, A^{(k)}\right\}$. $A^{(k)}$ is the winning matrix of our side in the k^{th} phase.The sequence of engagements can be abstracted as a sequential game model with m stages. So in this scenario, the demonstration data set is shown in (3).

$$D_E = \left\{\left(S_i, G^{(i)}\right)\right\}_{i=1}^{m} \tag{3}$$

B. IMITATION LEARNING WITH THE DECOUPLED NETWORK (ILDN)

1) DECOUPLED NETWORK FRAMEWORK

The process of IL is divided into two significant steps, sampling, and learning. In this paper, sampling is done through KRB, while learning is done by using samples to optimize neural network parameters, so the framework of DRL does not apply. This paper proposes a decoupled framework to decouple the network into two parts: inference and training. Inference part: KRB interacts with the environment and stores the samples into Replay Buffer.Training part: The neural network optimizes the parameters based on the samples and iterates repeatedly to finally make the decisions of the neural network similar to KRB. The decoupled network framework is shown in Fig. 1.

The essence of this framework is based on the idea of DRL training, combined with the requirements of IL, to solve the ground-air confrontation task assignment problem. In this framework, the training network needs to be trained, and KRB generates the sample data for training. In synchronous mode, the two processes, sampling and learning, have to wait for each other, significantly increasing the execution time. To speed up the work efficiency and efficiently use the limited computational resources, this paper adopts an asynchronous training mode based on the decoupled network framework, where the sampling process and the learning process are parallel without waiting for each other, and computational resources are reasonably allocated according to the sampling and learning demands. As shown in Fig. 2, the decoupled network framework in this paper can meet IL's needs for smoother training while maximizing computational resources and improving training efficiency.

2) PRE-TRAINING METHOD BASED ON BEHAVIORAL CLONING

We have completed modeling the MDP of this study in Section III.C by defining the four elements (S, A, r, P). The objective is to maximize the desired cumulative reward value by solving the optimal policy π^* with unknown $p\,(s_{t+1}|s_t, a_t)$

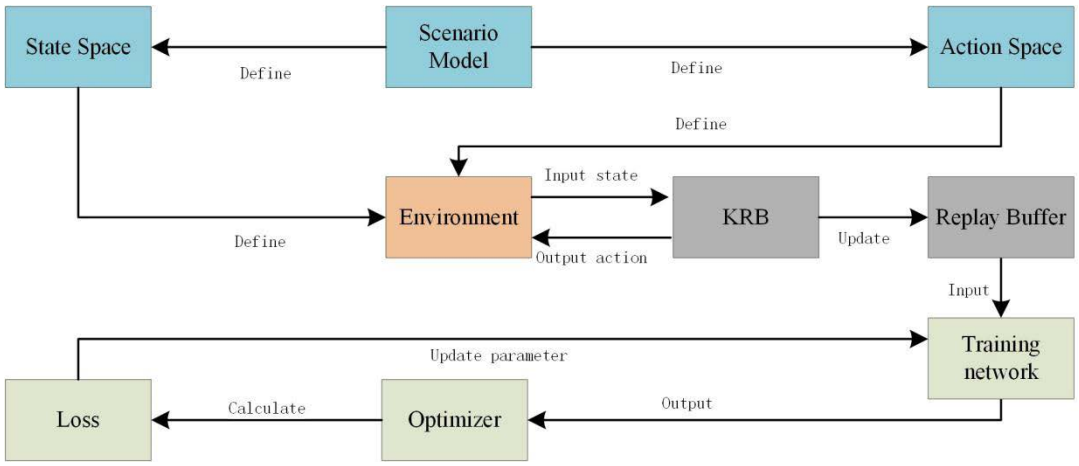

FIGURE 1. Decoupled network framework.

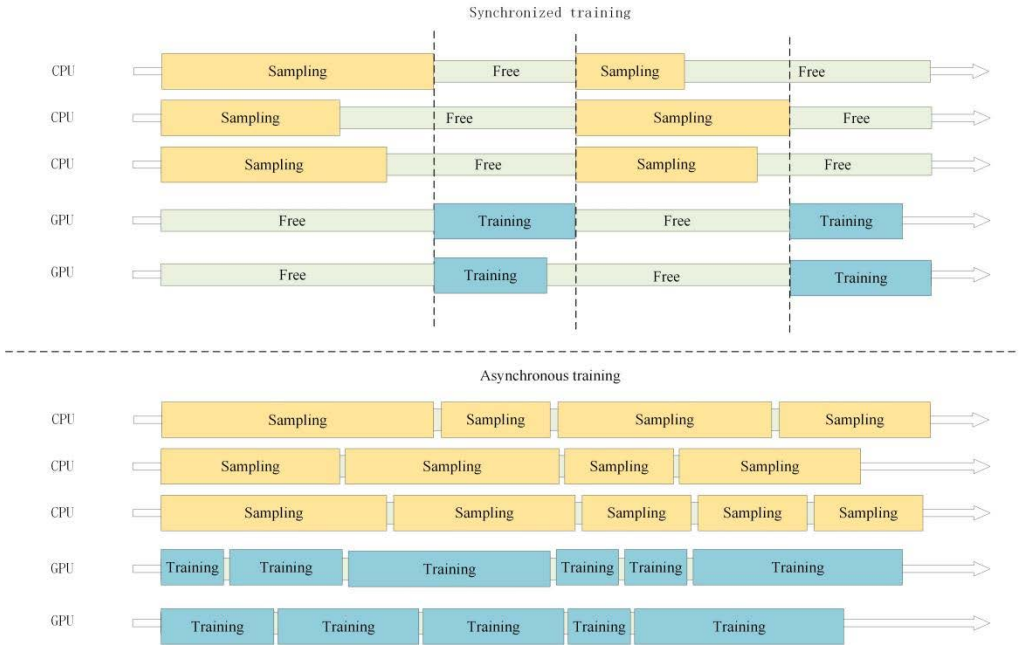

FIGURE 2. Comparison of the working process of the training mode.

using a reinforcement learning algorithm, as shown in (4).

$$\max_{\pi} E\left[\sum_{t=0}^{T} \gamma^t r_t\right] \quad \text{s.t. } s_{t+1} \sim p\left(\cdot|s_t, a_t\right), \quad a_t \sim \pi\left(\cdot|s_t\right), \ t = 0, \ldots T-1 \tag{4}$$

In the pre-training phase, we define the optimal policy π^* as the policy of KRB that needs to be learned. We constructed a Demonstration Buffer to deposit the expert data for this problem, drawing on the pre-training methods of DQNfD [6] and DDPGfD [27]. In the pre-training phase, the agent is trained by drawing small batches of samples from the Demonstration Buffer. The essence of this training approach is to find a strategy with the smallest difference in value function from the expert strategy [28], [29], as shown in (5).

$$\min_{\pi} \left[V\left(\pi_E\right) - V\left(\pi\right)\right] \tag{5}$$

For the agent to learn expert strategies stably, we added the calculation of Behavior Cloning Loss [30], as shown in (6).

$$L_{BC} = \sum_{i=1}^{\left(S_i, G^{(i)}\right)} \left\|\pi\left(s_i|\theta_\pi\right) - a_i\right\|^2 \tag{6}$$

Due to the need to improve the credibility of the agent's decision in this scenario, at this stage, we let the strategy that the agent eventually learns be exclusively an expert strategy. Therefore we minimize the difference between the actions of

the two strategies by L_{BC}, where $\left(S_i, G^{(i)}\right)$ is the sample set in Section A.

3) GATED RECURRENT UNIT

Gated Recurrent Unit (GRU) is used to retain data that needs to be remembered while also selectively forgetting unimportant information. GRU alleviates the problem of gradient disappearance compared to Recurrent Neural Network (RNN) and will train faster with fewer tensor operations compared to Long Short-Term Memory (LSTM) networks [31]. GRU combines the input gate with and forgetting gate to generate an update gate, while GRU directly defines a linear dependency between the current state h_t and the historical state h_{t-1}.

In GRU, the candidate state $\tilde{h}_t$ at a given moment is

$$\tilde{h}_t = \tanh\left(Wx_t + U\left(r_t \odot h_{t-1}\right) + b\right) \tag{7}$$

where $r_t \in [0, 1]$, defined as the reset gate, is used to determine whether the calculation of $\tilde{h}_t$ forms a dependency on the state h_{t-1} at the last moment and $r_t = \sigma\left(W_r x_t + U_r h_{t-1} + b_r\right)$, where W_r and U_r are the weight parameters and b_r is the bias parameter. When $r_t = 0$, the candidate state $\tilde{h}_t = \tanh\left(W_c x_t + b\right)$ is only related to the input x_t, and is independent of the previous state. When $r_t = 1$, the candidate state $\tilde{h}_t = \tanh\left(Wx_t + Uh_{t-1} + b\right)$ is related to the input state$x_t$and the previous state h_{t-1}.

The GRU's hidden state h_t is updated in the following way:

$$h_t = z_t \odot h_{t-1} + (1 - z_t) \odot h_{t-1} \tag{8}$$

where $z \in [0, 1]$, defined as the update gate, is used to determine whether the current state retains some information of the previous state and whether to update the candidate state information, and $Z_t = \sigma\left(W_z x_t + U_z h_{t-1} + b_z\right)$.

When $z_t = 0$, there is no linear relationship between the current state h_t and the previous state h_{t-1}. If there is also $z_t = 0, r = 1$, the GRU network degenerates to an ordinary recurrent network; if there is also $z_t = 0, r = 0$, the current state $\tilde{h}_t$ is only related to the current input x_t and has nothing to do with the previous state h_{t-1}. Fig. 3 shows the specific structure of the GRU.

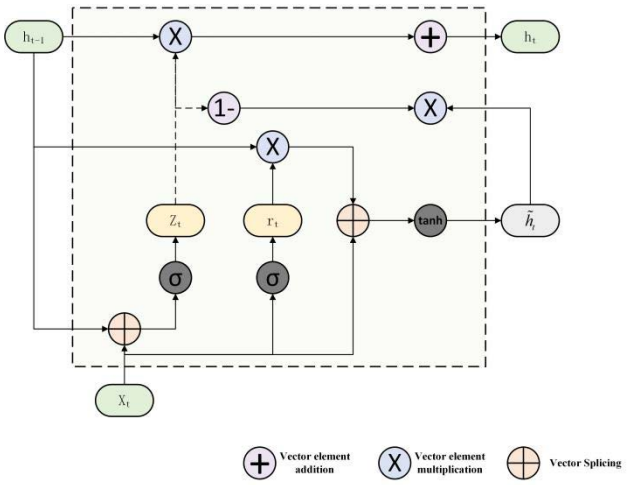

FIGURE 3. Structure of GRU.

C. SAFE REINFORCEMENT LEARNING FOR GROUND-TO-AIR CONFRONTATION(SRL-GC)

Since pre-training of IL often does not achieve optimal results, RL training after pre-training is also essential. The agent often performs unsafe explorations in RL training, such as letting high-threat targets approach and attacking high-value targets instead. Exploring these unsafe actions is not valuable for ground-to-air confrontation scenarios, and such exploration should be avoided as much as possible to improve the efficiency of later training. Safe RL can be a good balance between task performance and security [32]. A commonly used framework is to model Safe RL as a constrained Markov decision process (CMDP), which contains a cost function and the reward function of the MDP. The safety constraint is defined as the cumulative value of the cost function below a certain threshold.

Safe RL's CMDP model can be expressed as:

$$\begin{aligned} &\max_{\pi} \quad \mathop{\mathrm{E}}_{(s_t, a_t)\sim\rho_\pi}\left[\sum_t \gamma^t r\left(s_t, a_t\right)\right] \\ &s.t. \quad \mathop{\mathrm{E}}_{(s_t, a_t)\sim\rho_\pi}\left[\sum_t \gamma^t c\left(s_t, a_t\right)\right] \le d \end{aligned} \tag{9}$$

where $c\left(s_t, a_t\right)$ is the cost function and d is the cost threshold. $r\left(s_t, a_t\right)$ is the reward function. The goal of solving the problem is to maximize the long-term reward without exceeding a cost threshold constraint. Therefore, this is a constrained optimization problem, and a better compromise between reward and safety can be achieved by optimally satisfying the objective reward function under safety.

1) CONSTRAINT DESIGN FOR GROUND-TO-AIR CONFRONTATION TASK ASSIGNMENT

According to the requirements of the ground-air confrontation scenario, the constraints in this paper are designed as follows:

a: TRACKING FEASIBILITY CONSTRAINTS

Tracking feasibility is the basis for tracking task assignment, and for enabling stable tracking of the target, the constraint in (10) needs to be satisfied.

$$\begin{cases} H_{\min} \le H_T \le H_{\max} \\ \theta_{\min} \le \theta_T \le \theta_{\max} \\ D_{\min} \le D_T \le D_{\max} \\ \Delta_g < \Delta_{\max} \end{cases} \tag{10}$$

where, H_T, θ_T, D_T indicates the flight height, azimuth angle, and flight distance of the target. $H_{\min}, H_{\max}, \theta_{\min}, \theta_{\max}, D_{\min}, D_{\max}$ denotes the maximum value of the height, angle, and

distance illuminated by the sensor. Δ_g indicates the tracking error due to interference, and $\Delta_{\max}$ indicates the maximum allowable error of the sensor.

b: INTERCEPTION FEASIBILITY CONSTRAINT

Interception feasibility is a judgment condition for the missile to be able to destroy the target and must satisfy the constraint in (11).

$$\begin{cases} H'_{\min} \le H_T \le H'_{\max} \\ P_T \le P_{\max} \\ V_T \le V_{\max} \\ P'_{\min} < P'_T \end{cases} \tag{11}$$

where H_T, P_T, V_T, P'_T denotes the target flight altitude, course shortcut, flight speed, and predicted interception probability. $H'_{\min}$, $H'_{\max}$, $P_{\max}$, $V_{\max}$, $P'_{\min}$ denotes the minimum altitude, maximum value, maximum course shortcut, maximum motion speed, and minimum interception probability of missile interception, respectively.

c: RESOURCE CONSTRAINTS

The resource constraint is the main factor limiting the task allocation. The constraint of (12) needs to be satisfied when the mission access to different weapons needs to be reserved as much as possible to deal with unexpected situations while completing the mission.

$$\begin{cases} \sum_{T=1}^{n} x_{ij} \le G_i \\ \sum_{T=1}^{n} y_{ij} \le C_i \\ \sum_{i=1}^{m}\sum_{T=1}^{n} x_{ij} \le G, \sum_{i=1}^{m}\sum_{T=1}^{n} y_{ij} \le C \end{cases} \tag{12}$$

where G_i denotes the number of tracking task channels of interception unit U_i and C_i denotes the number of interception task channels of interception unit U_i. G, C denotes the total number of tracking and interception tasks that can be undertaken by all units, respectively.

2) PPO-LAGRANGIAN

The states that satisfy the above constraints are defined as the set of safe states S, and the rest are unsafe states. Then the cost function of this paper is shown in (13).

$$c(s_t) = \begin{cases} 1 & s_t \not\subset S \\ 0 & s_t \subset S \end{cases} \tag{13}$$

We equate the CMDP problem in this paper to an unconstrained max-min optimization problem based on the RL algorithm of the literature [25], combined with the PPO-Lagrangian algorithm [33] to solve.

$$\max_{\theta} \min_{\lambda \ge 0} \mathcal{L}(\theta, \lambda) \doteq r(\theta) - \lambda c(\theta) \tag{14}$$

In Equation (14), $r(\theta)$ is the reward function. $c(\theta)$ is the cost function, which is Equation (13) of this paper. λ is the Lagrangian operator, which is adaptively adjusted according to the training.

V. EXPERIMENTS AND RESULTS

A. CONFRONTATION SCENARIO SETTING

Taking the example of a red-side defence mission in a large-scale ground-to-air confrontation, the confrontation scenario refers to the literature [25]. Red set up seven long-range interceptor units and five short-range interceptor units to protect a command post and an airfield. The long-range interception unit consists of one long-range sensor and eight long-range interceptors, and the short-range interception unit consists of one short-range sensor and three short-range interceptors. Blue set up eighteen cruise bombs, twenty drones, twelve fighters, and two jammers to attack Red in batches. The above serves as a complex scenario for the experiments in this paper. The standard scenario set up in the experiment with five long-range interceptor units and three short-range interceptor units on the red side, ten cruise bombs, twelve UAVs, seven fighters, and two jammers on the blue side. The simple scenario is that the red side sets up three long-range interception units and three short-range interception units to defend one headquarters. The blue side sets up nine cruise bombs, eight UAVs, five fighters, and one jammer to attack the red side in two batches.

B. EXPERIMENTAL HARDWARE CONFIGURATION

The CPU running the simulation environment is an Intel Xeon E5-2678v3, with 88 core, 256 G memory; the GPU runs neural network training. The model is an NVIDIA GeForce 2080ti, with 72 cores and 11 G video memory. Where the CPU cluster is used for the sampling process of the DRL and the GPU cluster is used for the training process.

C. EXPERIMENT 1: VALIDATION OF THE EFFECT OF IL

The pre-trained agent with the ILDN method and KRB were put into the same scenario for 500 games offline inference, respectively, and the average battle damage comparison and behavior comparison of results are shown in Fig. 4 and Fig. 5.

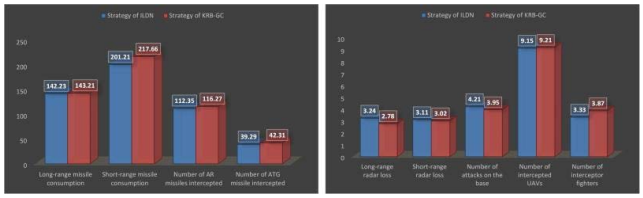

FIGURE 4. Average battle damage comparison.

The results of the battle damage comparison show that the decision-making level of the agent pre-trained by the ILDN is comparable to that of KRB. The behavioral analysis shows that the pre-trained agent (top right) can make decisions similar to the expert strategy (top left); for example, when

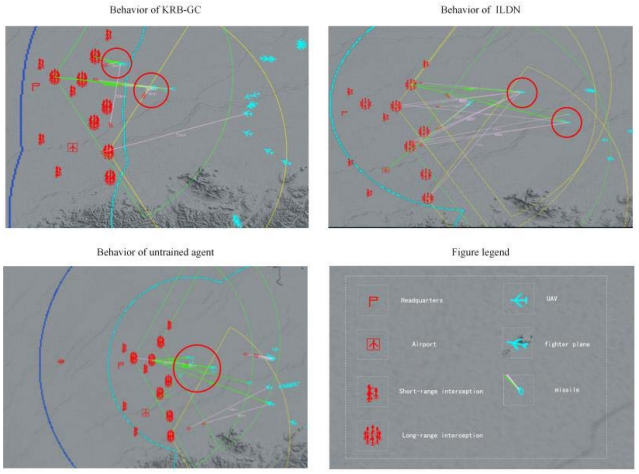

FIGURE 5. Comparison of behaviors in the same scenario.

Blue concentrates its fire on a particular interceptor unit, the surrounding interceptor units can share the pressure in time. The untrained agent (bottom left) uses an ineffective random strategy and has no sense of sharing the pressure to intercept. Experimentally, the agent pre-trained by the ILDN can learn a strategy that approximates KRB.

D. EXPERIMENT 2: VALIDATION OF THE EFFECT OF PRE-TRAINING

1) TRAINING DATA ANALYSIS

In three different scenarios, pre-trained agents by ILDN and that using ordinary DRL were iterated 50,000 times in the digital battlefield, respectively, and the comparison results are shown in Fig. 6.

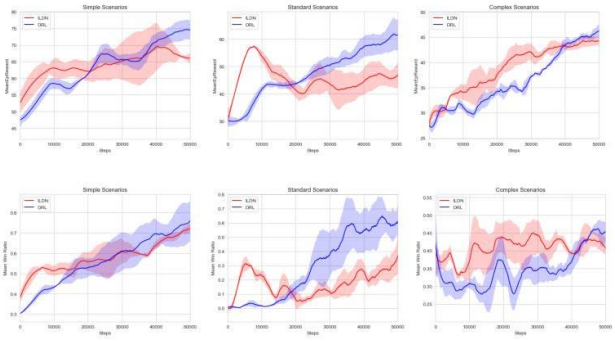

FIGURE 6. Comparison of training effects in different scenarios.

It can be seen that pre-training can make the agent reach a certain level quickly, and the more complex the scene, the more pronounced the effect of pre-training. However, the pre-trained agent does not perform consistently in subsequent RL training and is overtaken by the DRL agent as the number of iterations increases. Experimentally, in complex scenarios, ILDN can be good for improving the exploration efficiency in the pre-training stage of the agent. However, it is also needed to make the post-training more stable.

2) BEHAVIOR ANALYSIS

Pre-training aims to allow the agent to reduce ineffective exploration in the pre-training phase and quickly reach an approximate human level. So we trained LDN and DRL agents only 30,000 times in a complex scenario, performed behavioral analysis separately, and compared them with the untrained agent. The results of the comparison are shown in Fig. 7.

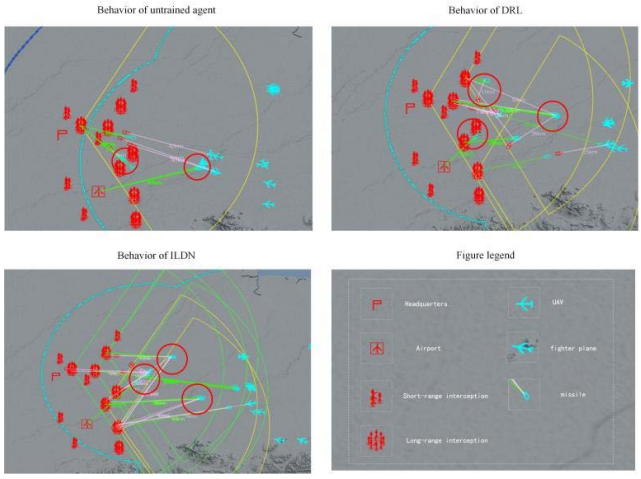

FIGURE 7. Comparison of behavioral details in complex scenarios.

It can be seen that the untrained agents (top left) use a random strategy of attacking only high-value targets instead of prioritizing the interception of high-threat targets; DRL agents (top right) have not yet learned a mature strategy at this stage and have an awareness of prioritizing the interception of high-threat targets, but the response timing is inaccurate; the agent that has been pre-trained by ILDN can already perform priority cooperative interception of high-threat targets at this stage. Experimentally, ILDN agents can learn mature strategies more quickly than DRL agents in complex scenarios.

E. EXPERIMENT 3: VALIDATION OF THE EFFECT OF SAFE RL

1) TRAINING DATA COMPARISON

We found that in complex scenarios, the pre-trained agent performed well in the pre-training period. Still, the results were not satisfactory in the post-training period. Therefore, we will verify the agent's performance in the later training period after adding the SRL-GC method. First, we train both the SITA method and the DRL method to full convergence in a standard scenario with less complexity to verify the effectiveness of the SITA method. The comparison results are shown in Fig. 8.

It can be seen that the SITA method converges faster than the DRL method and has higher reward values and win ratio after convergence.

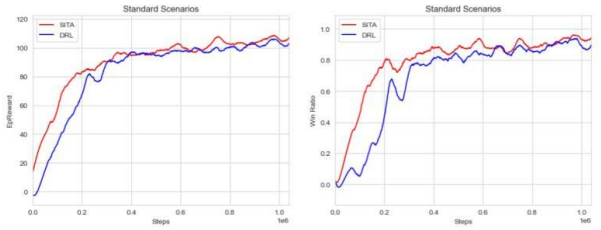

FIGURE 8. Comparison of training effects in standard scenarios.

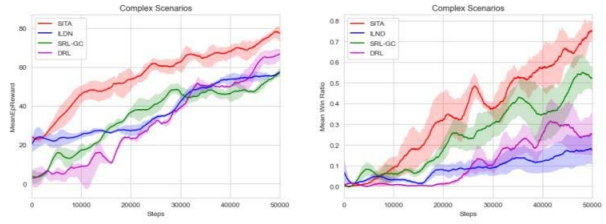

FIGURE 9. Comparison of training effects in complex scenarios.

Then, we trained SITA, ILDN, SRL-GC, and DRL in complex scenarios for 50,000 iterations to further validate the effectiveness of SITA. The comparison results are shown in Fig. 9.

It can be seen that DRL is very unstable in the early stage, the exploration efficiency is low, and the pre-training win rate is almost 0. ILDN can effectively improve the initial strategy level, and the initial average reward value can reach 20. Still, there is instability in the later training, and the reward value and win rate are difficult to improve. SRL-GC can effectively enhance the efficiency of intelligent body exploration. Still, the initial reward value and win rate are meager, which affects the overall. The SITA algorithm can effectively improve the post-training efficiency based on the pre-training strategy. The initial average reward value can also reach about 20. After 50,000 times of training, the final reward value can get about 80, and the last win rate is about 75%. Experimentally, in a complex environment, the SITA method is more efficient than the other three exploration methods and can obtain higher reward values and win rates at the same time step.

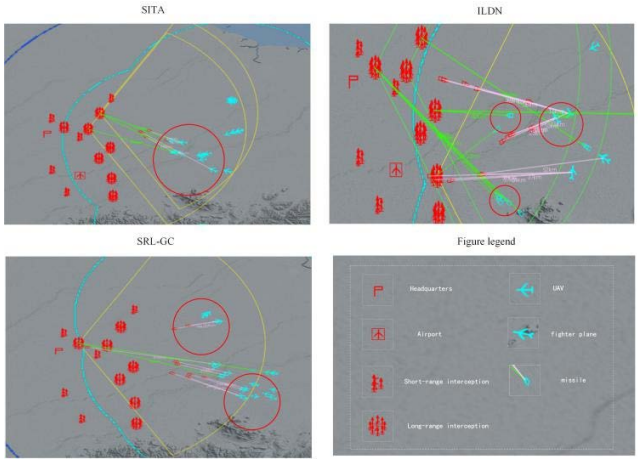

FIGURE 10. Comparison of the behavior of different agents in complex scenarios.

2) BEHAVIOR COMPARISON

As can be seen, when there is a mix of UAVs and human-crewed aircraft, the SITA can prioritize fire on manned units, and the ILDN can prioritize manned units but does not intercept highly threatening targets. The SRL-GC, on the other hand, does not have the sense to prioritize attacking

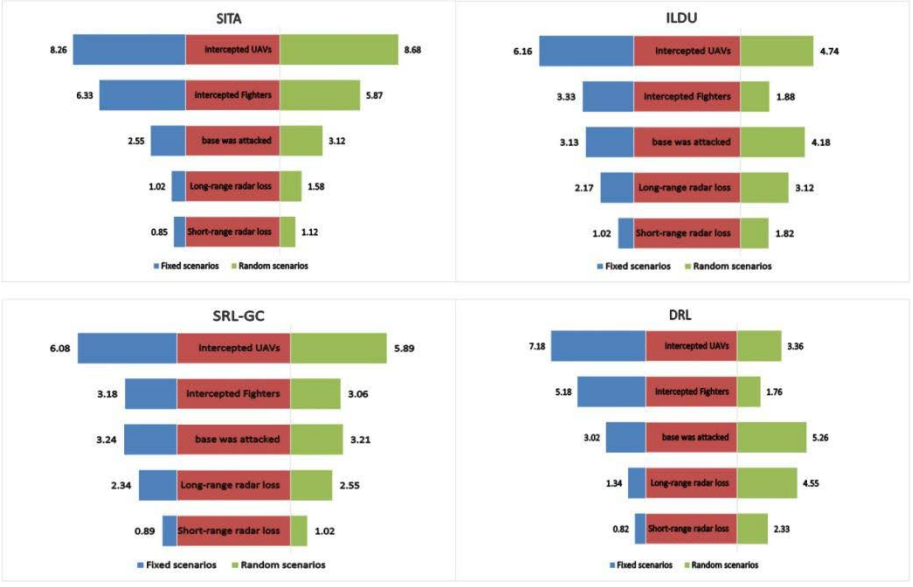

FIGURE 11. Comparison of the comparison results data.

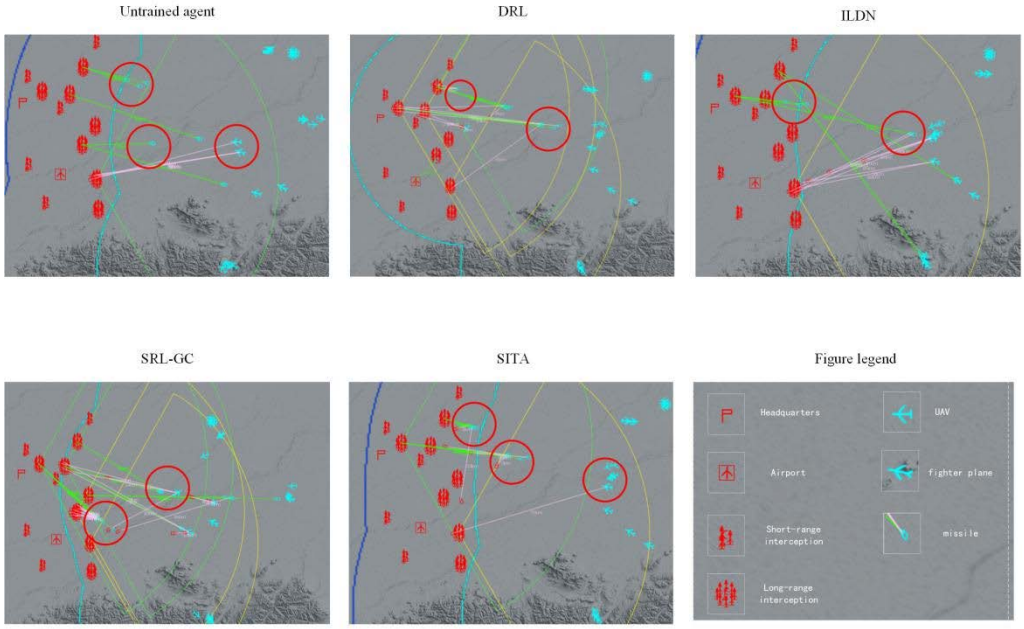

FIGURE 12. Comparison of the behavior of different agents in random scenarios.

high-value units at this stage, wasting too many resources. Experimentally, the strategy learned by the SITA method in the same time step is more reasonable than the other two methods.

F. EXPERIMENT 4:VERIFYING THE GENERALIZABILITY OF SITA

1) ANALYSIS OF CONFRONTATION RESULTS

In this experiment, we changed the fixed scenario of Experiment 3 by randomly transforming the assault route, arrival time, and detachment formation, with the incoming direction remaining unchanged overall. After training SITA, ILDN, SRL-GC, and DRL 50,000 times in the fixed scenario, they were confronted with the blue side for 500 games in both the fixed and random scenarios, and the data comparison results are shown in Fig. 11. The behavioral analysis results are shown in Fig. 12.

From the confrontation results and behavioral analysis, we can see that the change of scenario has a significant impact on DRL, which can no longer effectively respond to the blue attack, but the decision-making ability is still stronger than the untrained agent; the new scenario also has some impact on ILDN, but it still has an awareness of collaborative interception; SITA and SRL-GC are not sensitive to the change of scenario and can still effectively collaborate to intercept, but SITA still performs the best in the new scenario because of its higher training efficiency.

2) COMPARISON OF REWARD VALUE CHANGES

In this experiment, we put SITA, ILDN, SRL-GC, and DRL, which have been trained 5000 times in a fixed scene, in a random scenario to continue the training, and the results are shown in Fig. 13.

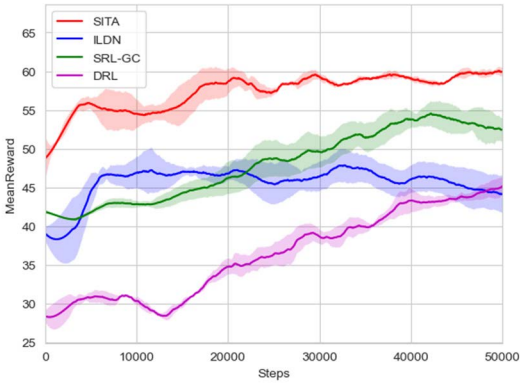

FIGURE 13. Comparison of training rewards for different agents in the new scenario.

Comparing Fig. 13 with the reward value curves in Fig. 9, it can be seen that the change in the scenario has some effect on the training of all four methods, and the decision level of all methods has decreased, with DRL decreasing the most significantly, with the average reward value dropping from 64 to 28. In the training of the new scenario, both ILDN and SRL-GC retained some of their original strategies, and both had initial average reward values of around 40. Still, they were less stable in the subsequent training.

SITA has the highest initial average reward value and is more stable in the subsequent training. Experimentally, the SITA method is more adaptive than the other three methods when the scenario is changed. However, the SITA method still has some limitations, the reward value is not significantly improved in the training after the scene change, and it is difficult to further improve the decision quality in the face

of new scenarios. In future work, we will address this point and further enhance the generalization ability of SITA.

VI. CONCLUSION

To address the problem of low training efficiency when DRL is applied to large-scale complex scenarios, this paper proposes the SITA method, which combines domain knowledge with DRL for task assignment in large-scale red-blue game confrontation scenarios. Firstly, a knowledge rule base is constructed based on the domain knowledge. The ILDN method is proposed to improve the training efficiency in the initial stage by using the demonstration samples. The SRL-GC method is combined to enhance the stability in the later stage of training. Finally, the effectiveness and superiority of the SITA method are verified on the digital battlefield. The experimental results show that the SITA method is more efficient than the standard DRL method in complex environments and can obtain higher reward values with the same number of training times. SITA also has a strong generalization capability to cope with changes in scenarios. Its strategy is more in line with the need for ground-to-air confrontation. It can provide new ideas and technical support for developing intelligent auxiliary decision-making systems.

In future work, we will continue to improve the generalization ability of SITA method to cope with more scenario changes, so that the method can better meet the actual needs of game confrontation scenarios.

DATA AVAILABILITY STATEMENT

The data presented in this study are available on request from the corresponding author. The data are not publicly available due to privacy reasons.

FUNDING AND CONFLICTS OF INTERESTS

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

REFERENCES

[1] B. Park, C. Kang, and J. Choi, "Cooperative multi-robot task allocation with reinforcement learning," *Appl. Sci.*, vol. 12, no. 1, p. 272, Dec. 2021, doi: 10.3390/app12010272.

[2] D. Wang and H. Deng, "Multirobot coordination with deep reinforcement learning in complex environments," *Expert Syst. Appl.*, vol. 180, Oct. 2021, Art. no. 115128, doi: 10.1016/j.eswa.2021.115128.

[3] R. N. Boute, J. Gijsbrechts, W. van Jaarsveld, and N. Vanvuchelen, "Deep reinforcement learning for inventory control: A roadmap," *Eur. J. Oper. Res.*, vol. 298, no. 2, pp. 401–412, 2022, doi: 10.1016/j.ejor.2021.07.016.

[4] R. Zhang, F. Torabi, G. Warnell, and P. Stone, "Recent advances in leveraging human guidance for sequential decision-making tasks," *Auto. Agents Multi-Agent Syst.*, vol. 35, no. 2, pp. 1–39, Oct. 2021, doi: 10.1007/s10458-021-09514-w.

[5] N. O. Lambert, D. S. Drew, J. Yaconelli, S. Levine, R. Calandra, and K. S. J. Pister, "Low-level control of a quadrotor with deep model-based reinforcement learning," *IEEE Robot. Autom. Lett.*, vol. 4, no. 4, pp. 4224–4230, Oct. 2019.

[6] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.

[7] V. Oriol, I. Babuschkin, and S. David, "Grandmaster level in StarCraft II using multi-agent reinforcement learning," *Nature*, vol. 575, no. 7782, pp. 350–354, 2019.

[8] H. Wang, S. Yuan, M. Guo, X. Li, and W. Lan, "A deep reinforcement learning-based approach for autonomous driving in highway on-ramp merge," *Proc. Inst. Mech. Eng., D, J. Automobile Eng.*, vol. 235, nos. 10–11, pp. 2726–2739, Sep. 2021, doi: 10.1177/0954407021999480.

[9] H. Ying, Z. Zheng, and Y. Richard, "Deep-reinforcement-learning-based optimization for cache-enabled opportunistic interference alignment wireless networks," *IEEE Trans. Veh. Technol.*, vol. 66, no. 11, pp. 10433–10445, Nov. 2017.

[10] T. V. Samak, C. V. Samak, and S. Kandhasamy, "Robust behavioral cloning for autonomous vehicles using end-to-end imitation learning," 2020, *arXiv:2010.04767*.

[11] H. Guo, Q. Chen, Q. Xia, and C. Kang, "Deep inverse reinforcement learning for objective function identification in bidding models," *IEEE Trans. Power Syst.*, vol. 36, no. 6, pp. 5684–5696, Nov. 2021, doi: 10.1109/TPWRS.2021.3076296.

[12] J.-L. Lin, K.-S. Hwang, H. Shi, and W. Pan, "An ensemble method for inverse reinforcement learning," *Inf. Sci.*, vol. 512, pp. 518–532, Feb. 2020.

[13] J. Schulman, S. Levine, P. Abbeel, M. Jordan, and P. Moritz, "Trust region policy optimization," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 1889–1897.

[14] M. Wen and U. Topcu, "Constrained cross-entropy method for safe reinforcement learning," *IEEE Trans. Autom. Control*, vol. 66, no. 7, pp. 3123–3137, Jul. 2021.

[15] H. Sun, Z. Peng, B. Dai, J. Guo, D. Lin, and B. Zhou, "Novel policy seeking with constrained optimization," 2020, *arXiv:2005.10696*.

[16] S. Gangapurwala, A. Mitchell, and I. Havoutis, "Guided constrained policy optimization for dynamic quadrupedal robot locomotion," *IEEE Robot. Autom. Lett.*, vol. 5, no. 2, pp. 3642–3649, Apr. 2020, doi: 10.1109/LRA.2020.2979656.

[17] Y. Chow, O. Nachum, A. Faust, M. Ghavamzadeh, and A. E. Duenez-Guzman, "Lyapunov-based safe policy optimization for continuous control," 2019, *arXiv:1901.10031*.

[18] H. Liu, J. Ge, Y. Wang, J. Li, K. Ding, Z. Zhang, Z. Guo, W. Li, and J. Lan, "Multi-UAV optimal mission assignment and path planning for disaster rescue using adaptive genetic algorithm and improved artificial bee colony method," *Actuators*, vol. 11, no. 1, p. 4, Dec. 2021, doi: 10.3390/act11010004.

[19] G. Attiya and Y. Hamam, "Task allocation for maximizing reliability of distributed systems: A simulated annealing approach," *J. Parallel Distrib. Comput.*, vol. 66, no. 10, pp. 1259–1266, Oct. 2006.

[20] J. Nalini and P. M. Khilar, "Reinforced ant colony optimization for fault tolerant task allocation in cloud environments," *Wireless Pers. Commun.*, vol. 121, no. 4, pp. 2441–2459, Dec. 2021.

[21] Z. Zeng, H. Bao, Z. Wen, and W. Zhu, "Object tracking using the particle filter optimised by the improved artificial fish swarm algorithm," *Int. J. Intell. Inf. Database Syst.*, vol. 12, nos. 1–2, pp. 6–19, 2019, doi: 10.1504/IJIIDS.2019.102323.

[22] K. Rajchandar, R. Baskaran, P. K. Padmanabhan, and M. Rajmohan, "A novel fuzzy and reverse auction-based algorithm for task allocation with optimal path cost in multi-robot systems," *Concurrency Comput., Pract. Exper.*, vol. 34, no. 5, pp. 1–15, 2021, doi: 10.1002/cpe.6716.

[23] J. Zhang, G. Wang, and Y. Song, "Task assignment of the improved contract net protocol under a multi-agent system," *Algorithms*, vol. 12, no. 4, p. 70, Apr. 2019.

[24] Y. Li, W. Han, and Y. Wang, "Deep reinforcement learning with application to air confrontation intelligent decision-making of manned/unmanned aerial vehicle cooperative system," *IEEE Access*, vol. 8, pp. 67887–67898, 2020, doi: 10.1109/ACCESS.2020.2985576.

[25] J.-Y. Liu, G. Wang, Q. Fu, S.-H. Yue, and S.-Y. Wang, "Task assignment in ground-to-air confrontation based on multiagent deep reinforcement learning," *Defence Technol.*, vol. 2022, pp. 1–10, Apr. 2022, doi: 10.1016/j.dt.2022.04.001.

[26] Q. Fu, C.-L. Fan, Y. Song, and X.-K. Guo, "Alpha C2—An intelligent air defense commander independent of human decision-making," *IEEE Access*, vol. 8, pp. 87504–87516, 2020, doi: 10.1109/ACCESS.2020.2993459.

[27] M. Vecerik, T. Hester, J. Scholz, F. Wang, O. Pietquin, B. Piot, N. Heess, T. Rothörl, T. Lampe, and M. Riedmiller, "Leveraging demonstrations for deep reinforcement learning on robotics problems with sparse rewards," 2017, *arXiv:1707.08817*.

[28] X. Tian, L. Ziniu, and Y. Yang, "Error bounds of imitating policies and environments for reinforcement learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 10, pp. 6968–6980, Oct. 2022.
[29] S. Ross, G. Gordon, and D. Bagnell, "A reduction of imitation learning and structured prediction to no-regret online learning," in *Proc. 14th Int. Conf. Artif. Intell. Statist.*, 2011, pp. 627–635.
[30] A. Nair, B. McGrew, M. Andrychowicz, W. Zaremba, and P. Abbeel, "Overcoming exploration in reinforcement learning with demonstrations," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2018, pp. 6292–6299.
[31] P. Tiwari, H. Zhu, and H. M. Pandey, "DAPath: Distance-aware knowledge graph reasoning based on deep reinforcement learning," *Neural Netw.*, vol. 135, pp. 1–12, Mar. 2021, doi: 10.1016/j.neunet.2020.11.012.
[32] J. García and D. Shafie, "Teaching a humanoid robot to walk faster through safe reinforcement learning," *Eng. Appl. Artif. Intell.*, vol. 88, Feb. 2020, Art. no. 103360, doi: 10.1016/j.engappai.2019.103360.
[33] Q. Liang, F. Que, and E. Modiano, "Accelerated primal-dual policy optimization for safe reinforcement learning," 2018, *arXiv:1802.06480*.

JIAYI LIU was born in Fuzhou, Fujian, China, in 1996. He received the bachelor's degree from Beijing Forestry University, in 2018. He is currently pursuing the Ph.D. degree with Air Force Engineering University. His research interests include deep reinforcement learning, intelligent decision aiding systems, and artificial intelligence.

GANG WANG was born in Qingzhou, Shandong, China, in 1975. He is currently a Professor at Air Force Engineering University. His research interests include intention recognition, intelligent information processing, and artificial intelligence.

XIANGKE GUO was born in Henan, China, in 1980. He received the M.S. degree from Air Force Engineering University, in 2007, and the Ph.D. degree from the School of Electronics and Information Engineering, Beihang University, in 2018. His current research interests include target tracking and intelligent information processing.

SIYUAN WANG was born in Taiyuan, Shanxi, China, in 1994. He received the bachelor's degree from the Beijing Institute of Technology, in 2017. He is currently pursuing the Ph.D. degree with Air Force Engineering University. His research interests include intention recognition, intelligent information processing, and artificial intelligence.

QIANG FU was born in Shaanxi, China, in 1988. He received the Ph.D. degree from Air Force Engineering University. He is currently a Lecturer with Air Force Engineering University. His research interests include intelligent information processing, intelligent control, and fuzzy sets.