

Article

HC-MARL: A General Hierarchical Cascaded Multi-Agent Collaborative Architecture for Cyber Wargaming Applications

Zhiqiang Qu ^{1,2} , Jun He ^{2,*}, Bo Wu ², Zhitao Long ² and Tao Xia ² ¹ National University of Defense Technology, Changsha 410073, China; quzhiqiang@nudt.edu.cn² National Defense Branch of the National Engineering Research Center of Information Content Analysis Technology, Information Support Force Engineering University, Wuhan 430000, China; cherry_wb@nudt.edu.cn (B.W.); longzhitao@alumni.nudt.edu.cn (Z.L.); xiatao17@nudt.edu.cn (T.X.)

* Correspondence: hejun17c@nudt.edu.cn

Abstract

Cyber wargaming serves as a core tool for simulating cyber confrontations and supporting operational decision verification. Existing multi-agent methods face issues such as the absence of cross-level mechanisms and poor adaptability to dynamic environments when applied to cyber wargaming. To address these challenges, we innovatively propose HC-MARL, a general hierarchical cascaded multi-agent reinforcement learning architecture tailored for cyber wargaming. Within this architecture, agents are modeled as hierarchical cascaded units to achieve structural decoupling and preserve scalability in both horizontal and vertical dimensions. Specifically, we devise a cross-level bidirectional information passing and intrusion alert sharing mechanism to accommodate cross-level command characteristics; design a Transformer-based message transformation function that fuses variable-length observation vectors from lower-level agents into fixed-length vectors, adapting to dynamic changes in agent structures and numbers; and design a policy function that integrates neural networks with empirical knowledge, incorporating an intrusion-alert-based action mask to enhance threat response efficiency. To the best of our knowledge, the proposed HC-MARL framework is a novel general hierarchical collaborative multi-agent architecture for cyber wargaming. To validate the effectiveness of our framework, we instantiate two-layer and three-layer agent clusters and conduct experiments in the CyBORG CC4 environment. The results demonstrate that our architecture can effectively accommodate cross-level information transfer and dynamic changes in agents in cyber wargaming scenarios. Furthermore, incorporating intrusion-alert-based action masking into the policy function significantly improves defensive performance. Compared with the state-of-the-art (SOTA) methods, the average reward improves by approximately 12.3%, and compared with mainstream methods such as Singh et al., the average reward improves by approximately 27.5%.



Academic Editor: Hung-Yu Chien

Received: 15 May 2026

Revised: 27 July 2026

Accepted: 28 July 2026

Published: 31 July 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

Keywords: HC-MARL; multi-agent reinforcement learning; hierarchical reinforcement learning; C2; cyber wargaming

1. Introduction

Cyberspace has emerged as the fifth operational domain, following land, sea, air, and space, and cyber confrontation has become a critical component of modern warfare. As a simulation tool that integrates offensive and defensive technologies, game theory, and operational coordination, cyber wargaming constructs virtual cyber confrontation scenarios

to effectively simulate the attack and defense actions, decision-making processes, and confrontation outcomes of red and blue teams, thereby providing crucial support for the formulation of cyber operation plans, operational capability assessment, and personnel training. In particular, cyber wargaming exercises can construct a set of hypothesized capabilities, tools, and vulnerabilities using open-source information. These assumptions allow participants to gain more direct insight into how an adversary would act if equipped with such postulated resources and capabilities and how the defending side should respond accordingly.

Against the backdrop of artificial intelligence empowering cyber offense and defense, it is foreseeable that cyber confrontation will increasingly become automated and intelligent in the near future. AI-in-the-loop constitutes a fundamental requirement for cyber wargaming, with agent-versus-agent confrontation and human–agent hybrid confrontation serving as the essential modes of such exercises. Currently, no publicly available intelligent cyber wargaming system has been reported. While other domains, such as land, naval, and air wargaming, have been effectively integrated with intelligent adversarial gaming to meet the demands of intelligent warfare, we argue that intelligence has already become a defining attribute of contemporary cyber offense and defense. Therefore, exploring and investigating intelligent cyber wargaming is of considerable significance.

In the process of cyber defense, operational actions often exhibit multi-task, multi-objective, and cross-level characteristics. These actions—such as network surveillance, threat analysis, incident response, and security hardening—may be executed by different operational units (or agents) and are bound by tight logical interdependencies and resource constraints. In essence, the intelligent offensive–defensive adversarial game in cyber wargaming constitutes a multi-agent cluster task planning and collaborative optimization problem.

In recent years, multi-agent reinforcement learning (MARL) has achieved a series of breakthroughs in autonomous cyber defense [1–8]. In particular, hierarchical multi-agent reinforcement learning (HMARL) has demonstrated considerable potential in addressing complex network environments, multi-task autonomous coordination, algorithmic stability, and convergence speed [9–13]. Currently, two main approaches to HMARL exist in autonomous cyber defense. The first approach stratifies the agents' policies; for instance, Singh et al. [9] categorized policies into a master policy and sub-policies. The second approach constructs a two-layer agent architecture, where the upper-layer agent is responsible for selection or planning and the lower-layer agent is responsible for concrete execution [10,11]. In the hierarchical method proposed by Tang et al. [10], agents in both layers share the same global observations, and the rewards are shared between the two layers. These rewards consist of the reward for upper-layer selection, the reward for lower-layer actions, and the cost of actions. However, these hierarchical architectures are not suitable for cyber wargaming for several reasons. First, they fail to capture the characteristics of hierarchical command, making effective bidirectional information transfer between levels impossible. Second, the rigid structure of current multi-agent frameworks cannot accommodate normal combat attrition and dynamic adjustments that occur during cyber wargaming exercises. Third, the inherent long-horizon and sparsity of shared rewards lead to instability in learning.

Large language models (LLMs) have also demonstrated substantial potential in autonomous cyber defense [14–16]. They further explored the fusion of LLMs and RL, aiming to leverage LLMs as enhancement tools for RL agents. These studies collectively highlight the notable advantages of LLMs in explainability and general reasoning capability. Nevertheless, the inherent issues of decision latency and hallucination in LLMs critically undermine their real-time performance and reliability in practical defense settings, posing a major bottleneck for this line of research.

To design a general hierarchical collaborative framework that accommodates the cross-level command characteristics of cyber wargaming and enables flexible and efficient multi-agent coordination, the following issues must be further addressed. First, at the hierarchical level, the upper layer should be capable of controlling (or influencing) the actions of the lower layer through commands, while the lower layer should provide feedback on the status and outcomes of its actions to the upper layer; therefore, a cross-level bidirectional information exchange mechanism needs to be devised. Second, the layers in the architecture represent distinct hierarchical levels, and the decision-making scope and observed information vary according to the level at which an agent resides. Third, the architecture should allow flexible horizontal and vertical expansion to accommodate diverse operational scenarios and personnel configurations.

We argue that both biological systems and human cyber defense activities address complex task allocation and coordination problems through flexible, multi-scale hierarchical organization and that future intelligent defense systems should more closely resemble an organized society of agents. In light of this, we draw on the TAG architecture [17] and design a general-purpose hierarchical collaborative multi-agent architecture tailored for intelligent cyber wargaming scenarios, as illustrated in Figure 1. The agent at layer $l + 1$ can be likened to the commander-in-chief of cyber defense; the two agents at layer l are regional commanders, each responsible for defense tasks in different network regions. They receive commands from the superior layer, feedback the task results to the superior, and, when necessary, mutually communicate alert information. The agents at layer $l - 1$ are responsible for the actual execution of actions; they likewise receive commands from the regional commanders, report the action outcomes back to them, and communicate alert information to one another as needed. Moreover, the number of agents at each layer is variable, enabling the construction of agent collectives with variable depth and variable width based on this architecture. Notably, this structure aligns well with the command system of cyber defense: each agent needs only to attend to its own responsibilities and optimize its individual reward, while simultaneously coordinating with sibling agents through the hierarchical layout to contribute to the overall reward.

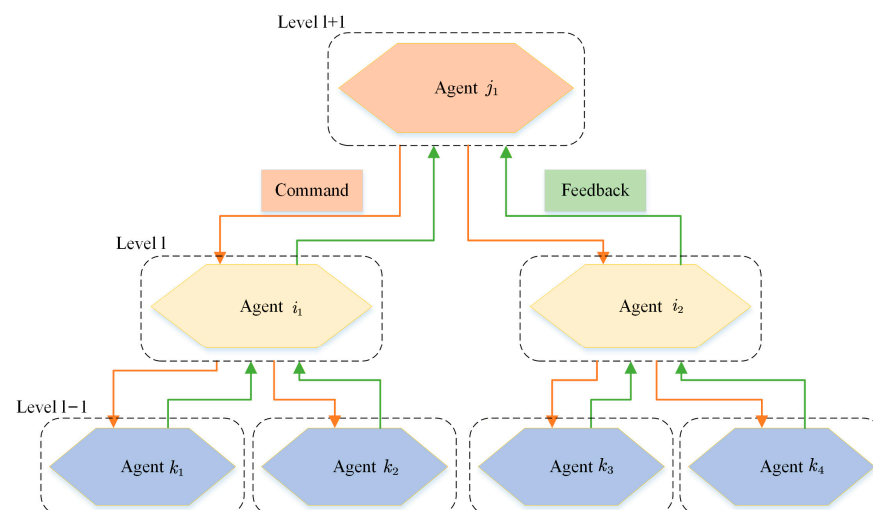


Figure 1. Diagram of the general hierarchical cascaded multi-agent cooperative architecture.

In the architecture illustrated in Figure 1, all agents share an identical structure. An agent influences its lower-level agents through commands, while feedback from the lower-level agents is transformed into its own observation vector. Regardless of the total number of layers in the system, each agent interacts solely with the immediately adjacent upper and lower layers, treating them as if they were part of the environment. Each layer can consist

of one or multiple agents, and the entire architecture can be arbitrarily scaled through horizontal and vertical cascading of agent units. This constitutes one of the key innovations of this work.

This approach ensures effective information transfer across multiple levels while preserving scalability through loose coupling between levels. To validate the effectiveness of the proposed method, we instantiated hierarchical structures with two-layer and three-layer agents, where each layer is responsible for making decisions at different levels and scopes. Experimental results demonstrate that the HC-MARL framework can be effectively applied to intelligent cyber wargaming; moreover, our method achieves significantly higher average rewards than both current SOTA methods and mainstream approaches. The main contributions are summarized as follows:

1. We propose HC-MARL (hierarchical cascaded multi-agent reinforcement learning), a general-purpose hierarchical collaborative multi-agent architecture for intelligent cyber wargaming, which consists of four bidirectional information flows for cross-level information passing and alert sharing. The architecture is inherently consistent with the organizational form of cyber defense.
2. We design a message transformation function that converts variable-length messages fed back from lower layers into fixed-length observation vectors, enabling adaptation to different network scenarios and personnel configurations.
3. We incorporate global intrusion alert information into the observation vector and design a policy function that integrates neural networks with alert-response mechanisms. Within this policy function, we embed an action-masking scheme based on intrusion alerts to enhance the efficiency of threat response.
4. Our proposed HC-MARL framework, while effectively adapting to cross-level command transmission and dynamic adjustments of agent structures, achieves significantly higher average rewards than current state-of-the-art methods.

The remainder of this paper is structured as follows. Section 2 reviews the theoretical background and contributions of prior studies. Section 3 elaborates on the design philosophy and methodology of the HC-MARL architecture, including the agent observation space, message transformation function, and policy function. Section 4 validates our method through experiments and analyzes the experimental results. Section 5 discusses its limitations and outlines future research directions.

2. Related Work

In recent years, cyber wargaming has played an increasingly prominent role in addressing cyber conflicts and cultivating professionals, with scholars worldwide conducting extensive research on its conceptual foundations, system architectures, and application scenarios. Fox et al. [18] focused on the adaptability of cyber wargaming in commercial settings, addressing the pain point that traditional wargames are predominantly oriented toward military and national security scenarios and are thus disconnected from actual business needs. They proposed an optimized wargaming framework for commercial scenarios. He et al. [19] targeted large-scale cyber confrontation scenarios, proposing a system architecture for wargaming based on large-scale cyber confrontation and conducting preliminary explorations of key technologies such as scenario reconstruction, offensive and defensive behavior modeling, and AI decision-making. Haggman [20] concentrated on the application of cyber wargaming in cybersecurity education, delving deeply into the design logic, implementation pathways, and effectiveness evaluation of educational wargames. It is a representative work in this niche area. Curry and Drage [21] analyzed the characteristics of different types of manual cyber wargames to help readers understand the categories of cyber wargames and meet their needs. Chen [22], through literature review, case analysis, and

expert interviews, examined the practicality of cyber wargaming in education and analysis, constructed the first standardized evaluation framework covering multiple types of cyber wargaming—analytical, educational, and commercial—and provided a critical analysis of several wargames. Roche [23] established the first complete theoretical system for cyber wargaming, clearly delineating the two core application scenarios of “research wargames” and “educational wargames,” and proposed the “wargaming trilemma” framework, which explicitly defines the trade-offs among analytical utility, situational realism, and player engagement, serving as a core evaluation criterion for subsequent wargaming design. Yang et al. [24] constructed a wargaming framework based on “composability + minimal operational units” and proposed a wargaming method tailored to the security evolution of power networks. Overall, existing research emphasizes the theoretical system and application scenario innovation, striving for technological breakthroughs and practical implementation, and has laid a certain foundation. However, there remains substantial room for improvement in adapting to future intelligent warfare and AI-empowered wargaming.

In essence, cyber wargaming is a game of adversarial confrontation between attackers and defenders within a given scenario, with the goal of enabling the wargaming party to take the most advantageous actions based on the offensive and defensive situation to maximize gains. Intelligent cyber wargaming, in which artificial intelligence (agents) assumes the role of the wargaming party, centers on solving core problems such as agent situational awareness, autonomous decision-making, and efficient collaboration. In recent years, deep reinforcement learning (DRL) and multi-agent reinforcement learning (MARL) have shown great potential in automated cyber defense. Papers [2,5,25,26] have successfully modeled cyber confrontations between blue and red agents, developing DRL techniques to train agents and producing highly capable autonomous defense agents. In particular, Kunz et al. [3] jointly trained red and blue agents, and experimental results showed that joint training could enhance the defensive capability of blue agents. However, such single-agent paradigms cannot meet the practical demands of hierarchical, region-specific, and multi-user wargaming. MARL leverages interactions and collaboration among multiple agents to detect, mitigate, and respond to cyber threats, offering a new avenue for intelligent cyber wargaming. Wiebe et al. [27] investigated the applicability of cooperative MARL in tactical-level cyber defense decision-making, comparing value-based independent learning and centralized training with decentralized execution (CTDE), and showed that both methods outperform heuristic defense methods. Oesch et al. [6] argued that agents should specialize in specific domains within the cyber defense process and conducted research on the role positioning, game mechanisms, environmental adaptability, and training environments of agents in cyber defense. In particular, hierarchical multi-agent reinforcement learning (HMARL) decomposes and stratifies defense tasks, with each sub-task assigned to a specialized agent. Singh et al. [9] proposed a hierarchical proximal policy optimization (PPO) architecture that decomposes cyber defense tasks into specific sub-tasks and divides the policy into a master policy and sub-policies, with sub-policies responsible for handling a category of sub-tasks and the master policy selecting which sub-task to execute. Hürten et al. [11] designed a hierarchical agent structure where a high-level agent coordinates the overall defense strategy and low-level agents focus on tasks in specific regions. Tang et al. [10] modeled cyber offense and defense as a Stackelberg hyper-game and used hierarchical MARL as the driving force for game evolution, proposing a two-layer agent architecture in which a first-level defense agent serves as a selector that activates a type of second-level defense agent, which then executes the actions. To address the issue that existing methods commonly use communication mechanisms or hybrid action spaces to cope with environmental non-stationarity and partial observability while relying on complex policy networks to process limited-bandwidth information, Wang

et al. [28] proposed a simplified cooperative multi-agent reinforcement learning framework that directly generates coordinated actions through customized communication protocols. Inspired by football games, Liu et al. [13] proposed a hierarchical commander-based policy optimization (HCPO) algorithm that deploys local commanders to retain the advantages of centralized training while circumventing communication constraints among agents. Castro et al. [14] conducted the first study on LLM-based agents in multi-agent autonomous cyber defense, integrating them with reinforcement learning (RL) agents within the CybORG CC4 environment and proposing a novel communication protocol for heterogeneous agent teams. Mukherjee et al. [15] proposed an LLM-based approach to automatically design reward functions for DRL agents in cyber defense. Ahmed et al. [16] proposed a hierarchical LLM-critic architecture for multi-agent cyber defense, where large language models serve as reward shapers and semantic feedback providers for reinforcement learning agents.

Furthermore, Palmer et al. [29] provided a review and critique of deep reinforcement learning (DRL) methods in autonomous cyber defense. Landolt et al. [4] reviewed recent advances in the application of multi-agent reinforcement learning (MARL) to automated cyber defense, with a particular focus on autonomous intelligent cyber defense agents and training environments, and outlined existing challenges and future research directions. Lazer et al. [30] examined the impact of artificial intelligence agents on cybersecurity, highlighting emerging threat models, security frameworks, and evaluation processes for agent systems, and analyzed systemic risks, including agent collusion, cascading failures, regulatory evasion, and memory poisoning.

Current research still exhibits the following deficiencies when applied to intelligent cyber wargaming: first, it fails to match the cross-level command characteristics inherent in wargaming and thus cannot achieve effective bidirectional information transfer between echelons. Second, existing multi-agent frameworks are structurally rigid and cannot accommodate normal battlefield attrition or dynamic structural adjustments during cyber wargaming.

3. Methods

3.1. Mathematical Representation of the Defense in Intelligent Cyber Wargaming

Multi-agent-based cyber wargaming exhibits the following core characteristics. First, it is inherently hierarchical, with clearly defined command and coordination relationships among agents operating at different levels. Second, it is highly dynamic: during adversarial engagement, the operational postures, resource states, and action strategies of both sides are in constant flux, requiring real-time adaptation by the agents. Third, the environment is partially observable: agents can only access local situational information and are unable to perceive the global picture, thereby necessitating information exchange to bridge the perceptual gap. Fourth, it entails multi-objective conflicts, in which the local objectives of individual agents may be misaligned, requiring collaborative mechanisms to achieve globally optimal outcomes.

Therefore, the coordination problem of defensive agents can be modeled as a decentralized partially observable Markov decision process (Dec-POMDP) [31]. A Dec-POMDP is a particular class of MDP in which multiple independent and decentralized agents with incomplete observations interact to optimize a shared reward signal. Formally, a Dec-POMDP is defined as a tuple $M = (N, S, A, T, O, \Phi, R, b_0)$, where $N = \{N_1, \dots, N_n\}$ is the set of n agents. S is a (finite) set of states, A is the set of joint actions, T is the transition probability function that specifies the probabilities $Pr(s'|s, a)$, O is the set of joint observations, Φ is the observation probability function that specifies the probabilities $Pr(o|a, s')$, R is the immediate reward function, and b_0 is the initial state distribution at time $t = 0$. In particular, $A = A_i \times A_j$ is the set of joint actions. Here, A_i is the set of actions available

to agent N_i . Similar to the set of joint actions, $O = O_i \times O_j$ is the set of joint observations, where O_i is a set of observations available to agent N_i .

At each time step t , each agent follows its local policy π_i to select an action $a \in A_i$ based on its observations and received information, which causes the environment to transition to a new state $s' \sim Pr(s'|s, a)$ and generates a global reward $r = R(s, a)$. The main goal for the blue agents is to maximize the expected sum of rewards, $TR = \sum_t r_t$.

3.2. HC-MARL Framework Design

To address the demands for cross-level command and multi-agent collaboration in cyber wargaming, this work proposes a general hierarchical cascaded multi-agent collaborative architecture. Through hierarchical decomposition, the responsibility boundaries of agents at each level are clearly delineated. Meanwhile, a top-down guidance information flow and a bottom-up feedback information flow are designed between levels to ensure effective bidirectional interaction. A global intrusion alert-based response mechanism is employed to improve the efficiency of threat response.

In this architecture, agents are organized in a tree-structured hierarchy according to command relationships. For a given agent, it may have multiple child nodes (i.e., command multiple operational units) but at most one parent node (the superior commander). Higher-level agents direct the lower-level ones, while the subordinates report execution results back to their parent node. Each agent makes decisions at the level that corresponds to its designated role, as shown in Figure 2.

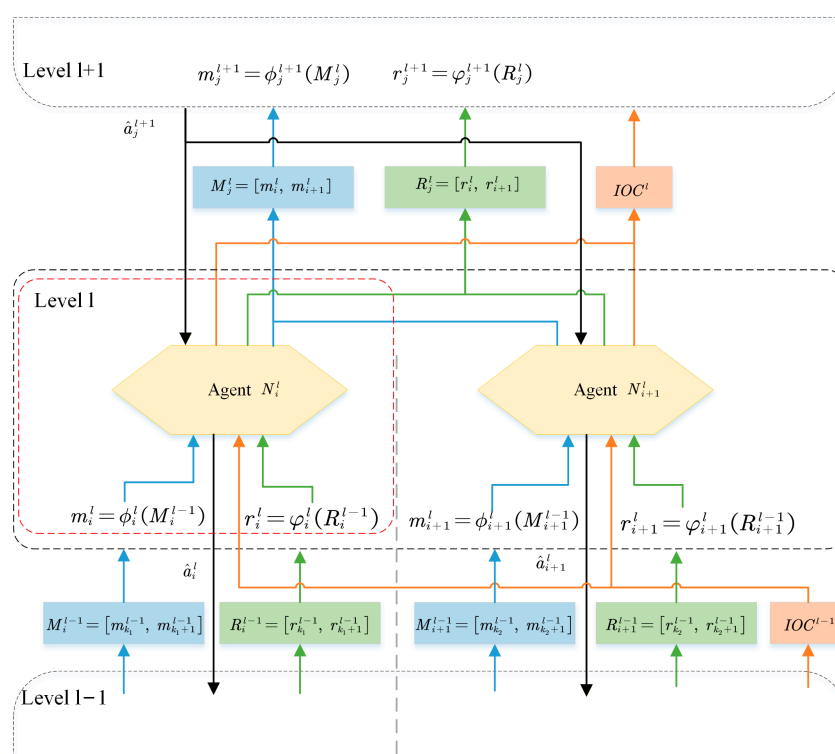


Figure 2. Diagram of the HC-MARL architecture and information flow.

Generally, for an L -layer multi-agent defense collective, the set of agents at layer l is denoted as $N^l = \{N_1^l, N_2^l, \dots, N_n^l\}$. In the hierarchical structure, each agent N_i^l is connected to agents in the immediately adjacent upper and lower layers. We denote the sets of agents at layer $l+1$ and $l-1$ as N^{l+1} and N^{l-1} respectively.

Similarly, the observation space of agent N_i^l is denoted as O_i^l , its action space as A_i^l , and its reward set as R_i^l . For notational convenience, we denote the parent agent of agent

N_i^l as $\hat{N}_i^l$ —in cyber wargaming, each agent has at most one parent at the upper level—and the set of child agents of N_i^l as $\hat{N}_i^l$.

In the HC-MARL architecture, four information flows exist between hierarchical levels. The parent node at the upper layer influences and shapes the observations of its child nodes through its chosen actions, forming a top-down guidance information flow. Conversely, the lower-layer agents transmit messages, rewards, and Indicators of Compromise (IOCs) to their superior agents, constituting a bottom-up feedback information flow. This design closely aligns with the operational process in which superior commanders issue orders to subordinate operators, while the latter report situational updates, emergent events, and action outcomes back to their superiors.

In cyber wargaming, operational scenarios and combat units can be highly dynamic, meaning that the number of agents at each layer may change over time. Existing MARL architectures are unable to accommodate this requirement. To address this challenge, we model agents as modular and reusable units, which we term hierarchical cascaded units (HCUs), as illustrated by the red dashed box in Figure 2. This implies that the observation spaces, action spaces, and the forms of policy and value functions across all hierarchical levels must remain consistent. A primary issue is thus the observation space: the number of subordinate child nodes is variable, which causes the lengths of the messages, rewards, and IOCs to vary with the number of child nodes. Additionally, the message formats between different levels need to be aligned.

To address this issue, we define a message transformation function $m_i^l = \Phi_i^l(M_i^{l-1})$ and a reward transformation function $r_i^l = \phi_i^l(R_i^{l-1})$ that convert the feedback from lower-level agents into fixed-length vectors, which then serve as part of the observation vector at the current level, where $M_i^{l-1} = [m_k^{l-1}]$, $N_k^{l-1} \in N_i^l$ and $R_i^{l-1} = [r_k^{l-1}]$, $N_k^{l-1} \in N_i^l$. The policy function is defined to determine the action to be taken, which can be expressed as $a_i^l = \pi_i^l(\hat{a}^{l+1}, m_i^l, IOC_i^l)$. Here, $\hat{a}^{l+1}$ incorporates the instruction action of the parent node, m_i^l is the transformed feedback messages from the child nodes, and IOC_i^l is the global indicator of compromise. In other words, agent N_i^l determines its optimal action based on the parent's instruction action, the feedback from its child nodes, and the global threat alert message. This design not only highlights hierarchy and modularity, ensuring efficient communication among agents, but also enhances vertical cascadability and horizontal flexible scalability. From the perspective of agent modeling, only a single HCU needs to be modeled, which greatly simplifies the modeling complexity.

3.3. Observation Space Design

To accommodate the requirements of cyber wargaming, the basic observation of the blue agent we design comprises four components. The first component indicates the current mission phase, which can take values in $\{0, 1, 2\}$. The second component represents the instruction from the superior commander, with a length of 8, encoding the priority levels of different actions. The third component captures the feedback information received from subordinate nodes (except for the lowest-layer agents, where the feedback is generated by the message transformation function Φ). The fourth component is the intrusion indicator, which indicates whether a global intrusion alert is present. In this paper, intrusion alerts mainly fall into three categories: the green agent's local work failed, discovery of the attacker's user privileges on a host, and discovery of the attacker's root privileges on a host. This is illustrated in Figure 3.

Mission	$\hat{a}^{l+1}$	m_i^l	IOC_i^l
---------	-----------------	---------	-----------

Figure 3. Composition of the agent observation space.

Note that the various components of the observation are presented to the agent in heterogeneous data formats, and these fields may vary across different agents, thus necessitating normalization of the observation.

3.4. Transformation Function Design

In the HC-MARL architecture, the message transformation function Φ and the reward transformation function φ serve to receive feedback information from a variable number of lower-level agents and fuse it into fixed-length vectors. To accommodate the interface between the lowest-layer agents and the network environment, the message vector is designed with a length of 76 in this paper. For the lowest layer, messages originate from the environment, whereas for all other layers, they are generated by the message transformation function Φ . Similarly, for the lowest layer, rewards come directly from the environment, while for other layers, they are produced by the reward transformation function φ . In this work, the reward transformation function employs a weighted average, while the message transformation function adopts a transformer model, the structure of which is illustrated in Figure 4.

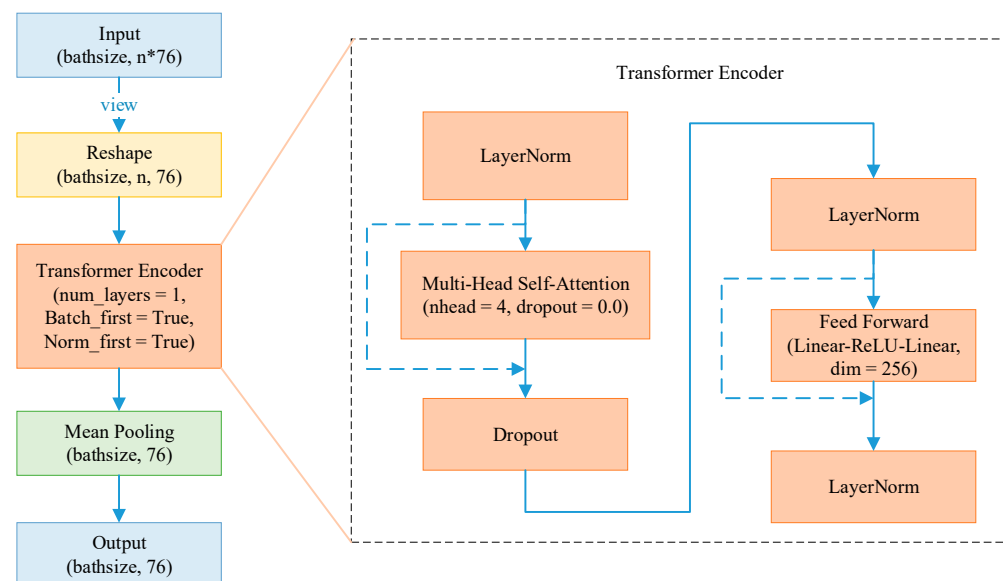


Figure 4. Network architecture of the message transformation function.

This model takes variable-length sequences as input and adopts the transformer encoder as its core feature extractor. Through an aggregation operation, it produces a fixed-length output. The advantage of designing the transformation function in this way is that it adapts to the dynamic changes in the number of agents at each layer while ensuring smooth information transfer, without requiring any modifications to the model structure. The hyperparameters are listed in Table 1.

Table 1. Hyperparameter settings of the transformer model.

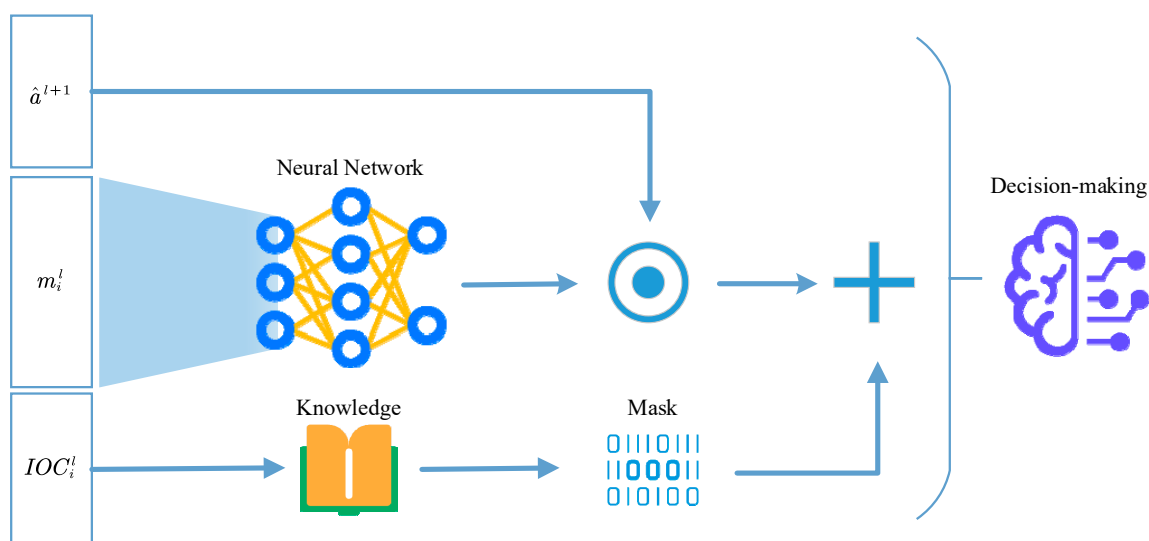
Hyperparameter	Values
d_model	76
nhead	4
num_encoder_layers	1
dim_feedforward	256
dropout	0.0
activation	ReLU

3.5. Policy Function Design

The policy function π takes the observation vector as input and outputs the selected action, i.e., $a_i^l = \pi_i^l(\hat{a}^{l+1}, m_i^l, IOC_i^l)$, where $\hat{a}^{l+1}$ incorporates the guiding information from the upper-layer parent node, m_i^l is the message received by the agent, and IOC_i^l is the global alert information. In this work, we design a policy function that integrates neural networks with empirical knowledge, as shown in Equation (1); the function f represents a neural network. The information processing pipeline is illustrated in Figure 5.

$$\begin{cases} \pi_i^l(\hat{a}^{l+1}, m_i^l, IOC_i^l) = \hat{a}^{l+1} \cdot f(m_i^l) + g(IOC_i^l) \\ g(IOC_i^l) = \max(\lg IOC_i^l, -10^6) \end{cases} \quad (1)$$

The neural network used here has 76 input nodes and 82 output nodes, employs the ReLU activation function, and contains two hidden layers of 256 nodes each. The message vector m_i^l serves as the input to the neural network, which outputs an action weight vector. This weight vector is then element-wise multiplied by the guidance information $\hat{a}^{l+1}$, which is expanded to 82 dimensions according to the action categories. Additionally, empirical knowledge is incorporated into the policy function along with a timely alert-response mechanism, whereby the IOCs are mapped to action masks. For instance, when a green agent suffers local work failure, a restore operation should be immediately triggered; if the attacker is found to have gained root or user privileges on a host, either removal or restoration must be promptly executed to curb threat escalation. In practice, if all IOCs are zero, all action masks are assigned a value of one. If any IOC is nonzero, the masks corresponding to the nonzero entries are set to one, and all others to zero. As a result, actions with a zero mask are effectively prohibited from being selected by the agent.

**Figure 5.** Information flow diagram of the policy function.

3.6. Learning and Optimization

In the HC-MARL architecture, each agent is required to learn two key functions: the policy function π for generating actions and the message transformation function Φ .

$$a_i^l = \pi_i^l(\hat{a}^{l+1}, m_i^l, IOC_i^l) \quad (2)$$

$$m_i^l = \Phi_i^l(M_i^{l-1}) \quad (3)$$

The policy function is responsible for mapping the combination of received guidance information and observations to the action of the current agent, while the message transformation function converts the feedback messages from lower-level agents into a fixed-length vector. The modular design of the framework allows agents at each level to learn independently using algorithms suited to their specific roles. This flexibility accommodates a wide range of learning methods. In this work, both the actor and critic networks are implemented as feedforward neural networks with two hidden layers, each containing 256 neurons. The policy function and the message transformation function are trained jointly, with policy optimization performed using the MAPPO algorithm. Since all agents are trained simultaneously, the action $\hat{a}^{l+1}$ employs the action weights of the superior agent from the previous time step.

During training, each agent stores its own experiences. The advantage estimates are then computed centrally using the GAE function, and each agent updates its policy and value networks accordingly based on the computed advantages, as outlined in Algorithm 1.

Algorithm 1 Learning algorithm

step()

 Input: $\hat{a}^{l+1}$

$a_i^l = \pi_i^l(\hat{a}^{l+1}, m_i^l, IOC_i^l)$

 for agent $N_i^l \in N^l$ do

$m_k^{l-1}, r_k^{l-1}, IOC_k^{l-1} \leftarrow \text{step}(a_i^l)$

 end for

$M_i^{l-1} = [m_k^{l-1}], N_k^{l-1} \in N^l, R_i^{l-1} = [r_k^{l-1}], N_k^{l-1} \in N^l$

$m_i^l = \Phi_i^l(M_i^{l-1}), r_i^l = \phi_i^l(R_i^{l-1}), IOC_i^l = \cup IOC_k^{l-1}, N_k^{l-1} \in N_i^l$

 if training then

 store($\hat{a}^{l+1}, m_i^l, IOC_i^l, a_i^l, r_i^l$)

 update()

 end if

 Return: m_i^l, r_i^l, IOC_i^l

4. Experiments

To validate the effectiveness of the proposed method, we conducted experimental evaluations in the CybORG environment of CAGE Challenge 4, which is the latest and most prominent open-source platform for autonomous cyber defense. This foundational platform provides simulation of the cyber behavior of attackers and defenders within realistic network topologies, enabling researchers to repeatedly construct, initialize, and test multi-agent models across a variety of cybersecurity scenarios. The original observation spaces, action spaces, and reward structures of the various agents are documented in detail, along with the source code, in the corresponding GitHub repository [32].

The primary hardware and software used in our experiments included: NVIDIA RTX3090 GPU, Ubuntu 22.04, Python 3.10.18, CUDA 12.1, CybORG v4.0, Gymnasium 0.28.1, Ray 2.10.0, and PyTorch 2.2.0.

We instantiated two-layer and three-layer multi-agent defense clusters in the CybORG CC4 network environment, illustrated in Figure 6. In the two-layer architecture, the top layer consists of a single agent that serves as the overall commander-in-chief of the defense, while the second layer comprises seven agents, each responsible for defending a distinct subnet. In the three-layer architecture, the top layer again consists of a single agent acting as the overall commander; the second layer contains three agents, which partition the seven subnets into three groups according to their functionality, with each agent responsible only for its assigned group; and the bottom layer is composed of seven agents, each tasked with executing specific operations—such as monitoring, analysis, deception, and removal—within its designated subnet.

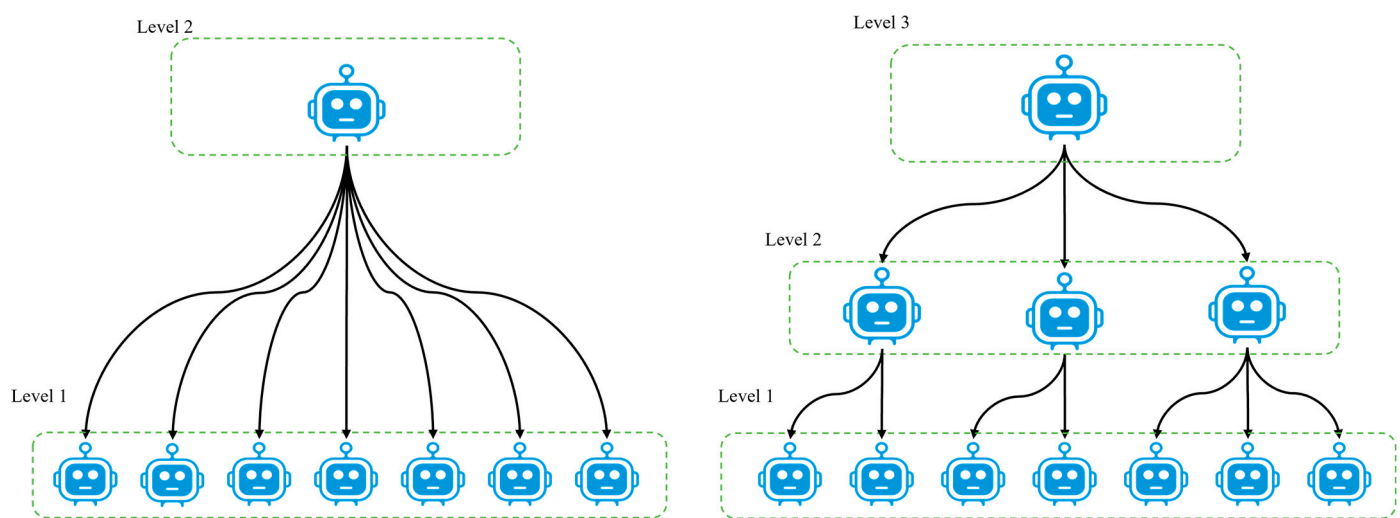


Figure 6. Hierarchical architecture of defensive agents.

For convenience of presentation, we denoted the method with a two-layer architecture as HC-MARL(2L), and the method with a three-layer architecture as HC-MARL(3L). To verify the generality and flexibility of the architecture, the number of agents within each subnet was configured heterogeneously.

In our experiments, we employed the MAPPO algorithm to update the policies (using the Ray RLlib library). The hyperparameters used during training were set as follows: learning rate of the agents in the first, second, and third layers of 3×10^{-4} , 1×10^{-4} , and 5×10^{-5} respectively; discount factor of 0.99; train batch size of 32,768; SGD minibatch size of 8192; num_sgd_iter of 15; and clip_param of 0.3. To guarantee fairness in comparison, we used the same training episodes and time steps per episode as in the other approaches. The training environment was set with a maximum of 500 time steps, and a total of 200 training iterations were conducted. After each training iteration, we evaluated the model with the same maximum time step length of 500, repeated over 10 times, and took the average reward and standard deviation over the 10 times as the test results. We trained and tested the two-layer and three-layer agent clusters using the same algorithm and hyperparameters, and the results are shown in Figure 7.

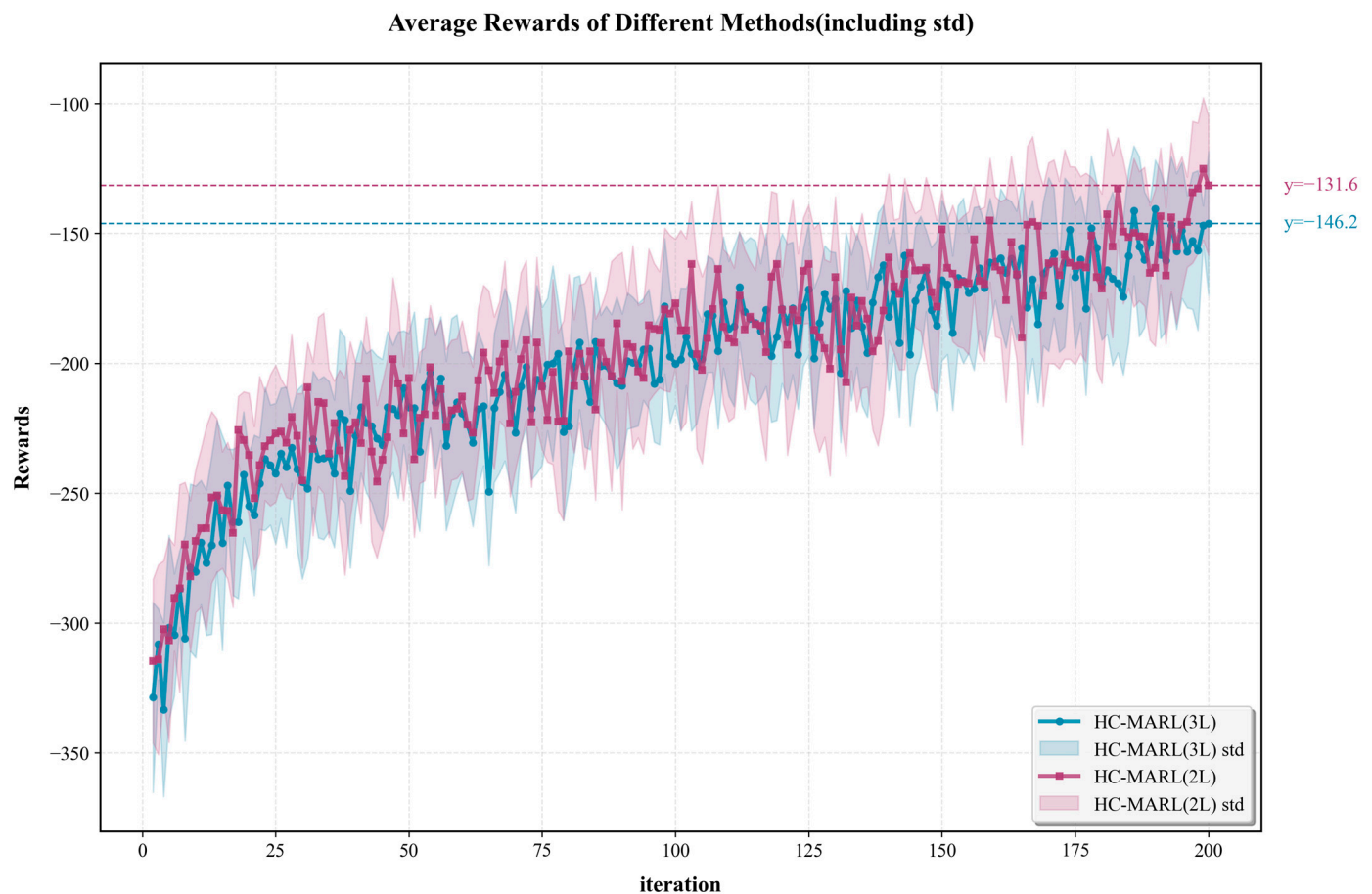


Figure 7. Comparison of different methods.

In Figure 7, the x -axis represents the training episodes, and the y -axis represents the average reward. The blue solid line shows the average reward of the HC-MARL(3L) method over 10 repeated tests, with the blue shaded area indicating the corresponding standard deviation. Similarly, the red solid line shows the average reward of the HC-MARL(2L) method, and the red shaded area indicates its standard deviation. From the figure, it can be observed that as the training episodes increase, the average reward gradually rises and then levels off. At the 200th episode, the average reward of the HC-MARL(3L) method is 146.2 with a standard deviation of 27.65, while that of the HC-MARL(2L) method is 131.6 with a standard deviation of 26.83.

The HC-MARL is a general multi-agent collaborative architecture that we specifically designed for cyber wargaming. First, we need to verify whether the decisions of the top-level agents can be effectively transmitted down to the bottom level—that is, whether the action selections of the bottom-level agents are significantly influenced by the top-level agents. In this work, we conducted comparative experiments on the three-layer agent cluster. By varying the guidance vector of the top-level agent, we counted the selection frequencies of actions taken by the bottom-level agents to observe the correlation between the two. In Experiment 1, we set the guidance vector of the top-level agent to $[0.125, 0.125, 0.125, 0.125, 0.125, 0.125, 0.125, 0.125]$, where all action types were assigned equal weights with no preferential tendency, serving as a baseline for the other experiments. In Experiment 2, we set the guidance vector to $[0.65, 0.05, 0.05, 0.05, 0.05, 0.05, 0.05, 0.05]$, indicating that the top level prefers Analyze actions. In Experiment 3, we set it to $[0.05, 0.05, 0.65, 0.05, 0.05, 0.05, 0.05, 0.05]$, where the top level prefers Remove actions. In Experiment 4, we set it to $[0.05, 0.05, 0.05, 0.05, 0.05, 0.05, 0.05, 0.65]$, indicating a preference for DeployDecoy actions. We conducted 10 repeated trials for each setting and recorded

the mean and standard deviation of the action types selected by the seven bottom-level agents. The results are presented in Table 2, where the first row denotes the action preference of the top-level agent, and the first column lists the available action types for the bottom-level agents.

Table 2. Impact of upper-level action preferences on lower-level action choices.

	No Preference	Analyze	Remove	DeployDecoy
Analyze	332 ± 21.06	* 339.7 ± 25.36	337.8 ± 20.7	321.1 ± 11.03
Remove	340 ± 22.11	342.2 ± 17.96	* 348 ± 20.52	342.4 ± 22.76
Restore	232.5 ± 13.75	227.4 ± 11.77	228.6 ± 15.56	231.1 ± 14.04
Sleep	2265.4 ± 18.41	2259.3 ± 16.14	2264 ± 16.97	2264.6 ± 11.69
AllowTraffic	0	0	0	0
BlockTraffic	0	0	0	0
DeployDecoy	315.4 ± 28.37	317.4 ± 15.15	307.6 ± 14.78	* 326.8 ± 15.69

* indicates that the actions of low-level agents align with the action preferences of top-level agents.

As shown in Table 2, when the top-level agent prefers to choose the Analyze action, the probability that the bottom-level agents select Analyze increases significantly, and the cumulative average of this action rises from 332 to 339.7. Similarly, when the top-level agent favors Remove, the selection probability of Remove among the bottom-level agents also increases markedly, and the cumulative average of this action rises from 340 to 348. When the top-level agent prefers DeployDecoy, the bottom-level agents show a significantly higher probability of choosing DeployDecoy, and the cumulative average of this action rises from 315.4 to 326.8. These experimental results demonstrate that, with the HC-MARL architecture, the decisions of the top-level agents can be effectively transmitted to the bottom level.

Second, we need to verify whether the HC-MARL framework can adapt to dynamic changes in agents within cyber wargaming. We conducted experiments on a three-layer agent cluster. We first trained the agents using the three-layer structure shown in Figure 8a and then performed various modifications to the cluster—such as adjusting the superior–subordinate relationships among agents, adding agents, and removing agents—to examine their impact on the cluster and the resulting performance.

We conducted four sets of comparative experiments. In Experiment 5, we only changed the superior–subordinate relationships among agents, adjusting agent N_2^1 to be a subordinate of agent N_2^2 , as shown in Figure 8b. In Experiment 6, we added an agent N_4^2 to the second layer, directly copied the policy of agent N_3^2 , and adjusted agent N_7^1 in the first layer to be its subordinate, as shown in Figure 8c. In Experiment 7, we removed agent N_2^2 and adjusted agents N_3^1 and N_4^1 in the first layer to be subordinates of agent N_1^1 , as shown in Figure 8d. For each experiment, we ran 10 test episodes and recorded the average reward and standard deviation over the 10 episodes; the results are presented in Table 3.

Table 3. The impact of agent structural adjustments on cluster performance.

	Agents (Level 1)	Agents (Level 2)	Agents (Level 3)	Rewards (<i>mean</i> ± <i>std</i>)
Original	7	3	1	−146.2 ± 27.65
Experiment 5	7	3	1	−154.8 ± 26.74
Experiment 6	7	4	1	−157.3 ± 27.31
Experiment 7	7	2	1	−153.7 ± 20.84

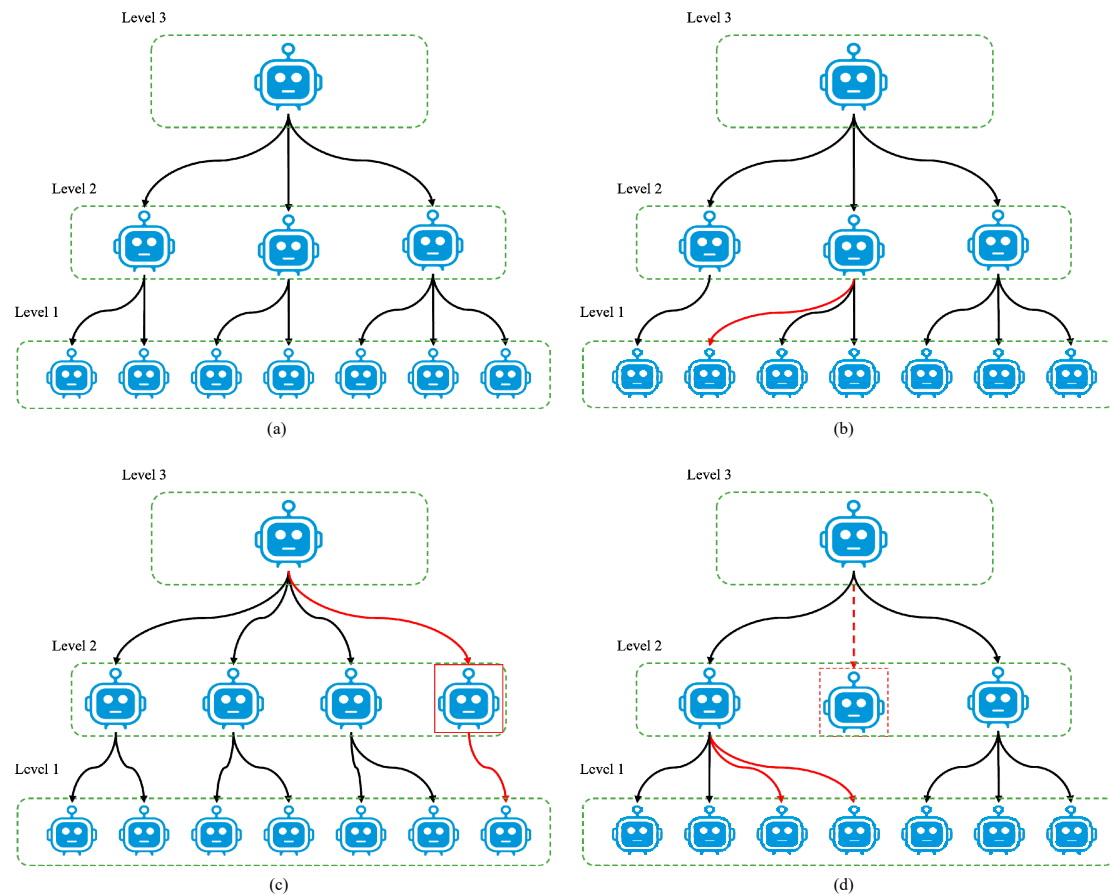


Figure 8. Change the structure of the agent cluster. (a) Original three-layer agent structure used; (b) Hierarchical relationships among agents changed; (c) New agents added; (d) Agents deleted.

As shown in Table 3, regardless of whether we adjust the superior–subordinate relationships among agents, add agents, or remove agents, the agent cluster can still function properly on the basis of the original trained model and maintain a satisfactory reward level. The largest variation occurs when adding agents, where the average reward decreases from -146.2 to -157.3 . These experimental results demonstrate that the HC-MARL framework can effectively adapt to dynamic changes in the agent cluster.

In this work, we designed a global intrusion alert IOC in the observation space and incorporated IOC-based action-masking information into the policy function. To validate its effectiveness, we conducted comparative experiments. We adopted a three-layer agent structure, selected the model trained for 200 episodes for testing, and performed 10 repeated test runs. The average reward and variance over the 10 runs are shown in Table 4.

Table 4. Impact of IOC-based action masking on rewards.

Methods	No IOC Mask	IOC Mask
Rewards (<i>mean ± std</i>)	-388.5 ± 13.86	-146.2 ± 27.65

It can be observed from Table 4 that, with the IOC-based action masking integrated into the policy function, the average reward rises from -388.5 to -146.2 . These experimental findings indicate that the IOC-driven emergency response mechanism plays a vital role in enhancing the reward, and this also manifests the integration of experiential knowledge with neural network models.

Finally, we compared the best reward achieved by the HC-MARL method with mainstream methods of Cybermonics [1], Singh [9], and ACDZero [33]. The results are shown

in Table 5. In the table, HC-MARL(3L) denotes the HC-MARL method with a three-layer agent structure, while HC-MARL(2L) denotes the HC-MARL method with a two-layer agent structure.

Table 5. The best rewards of different methods.

Methods	Cybermonic	Singh	ACDZero	HC-MARL(3L)	HC-MARL(2L)
Rewards (mean $\pm$ std)	-193 ± 84	-181.62 ± 65.85	-150.03 ± 19.85	-146.2 ± 27.65	-131.6 ± 26.83

As can be seen from the experimental data in Table 5, compared with the Cybermonic method, HC-MARL(3L) improves the average reward by 46.8 (24.2%) and increases the standard deviation by 56.35; compared with the Singh method, our proposed HC-MARL(3L) method achieves an average reward improvement of 35.42 (19.5%) and a standard deviation increase of 38.2; and compared with the ACDZero method, HC-MARL(3L) improves the average reward by 3.83 (2.5%) while reducing the standard deviation by 7.8. In particular, compared with the current state-of-the-art method (ACDZero), our proposed HC-MARL(2L) method achieves an average reward improvement of 18.43, corresponding to a relative increase of 12.3%; compared with mainstream methods such as Singh et al., the average reward improves by approximately 27.5%.

5. Discussion

In this work, we propose HC-MARL, a general hierarchical cascaded multi-agent framework that integrates cross-level command and dynamic node adjustment to overcome the limitations of existing MARL methods in intelligent cyber wargaming. Experimental results demonstrate that the framework exhibits the following characteristics:

- (1) The bidirectional information flow design of the HC-MARL framework not only ensures effective information transmission within the multi-agent system but also clarifies the core responsibilities and interaction boundaries among agents, thereby adapting well to the information activities between superior and subordinate levels in intelligent cyber wargaming.
- (2) The agent cluster structure based on the HC-MARL framework is highly flexible; through the design of message transformation functions, it can readily accommodate varying numbers of subordinate nodes, and the nodes can change dynamically.
- (3) The HC-MARL framework shows strong environmental adaptability; it supports multiple hierarchical cascade units that can adapt to different network environments and organizational structures through cascading.
- (4) While accommodating hierarchical information interaction and dynamic node changes, the general HC-MARL framework still achieves favorable defensive performance. Compared with the state-of-the-art methods, the HC-MARL(3L) method achieves an average reward improvement of 2.5%, and the HC-MARL(2L) method achieves an improvement of 12.3%.

The research outcomes presented in this paper provide new ideas and methodologies for multi-agent cyber wargaming, but there are still some directions that could be further improved:

- (1) The current method does not fully consider agent heterogeneity (e.g., capability differences among tactical agents of different types). Future work may incorporate heterogeneous multi-agent systems to optimize cooperative strategies for agents with diverse capabilities.

- (2) This work involves multiple neural network components. While some hyperparameter settings were borrowed from prior research, systematic comparative experiments on network architectures (e.g., the message transformation function) and hyperparameter configurations have not been performed, leaving room for more thorough tuning.
- (3) The use of global rewards and long-term reward mechanisms may lead to instability in the learning process of defensive agent policies. The integration of global rewards with local rewards remains an open issue that merits further exploration in future work.
- (4) Although the HC-MARL framework supports horizontal and vertical scaling, optimization becomes increasingly difficult as the structure grows more complex. The training methodology for multi-level cascaded policies thus warrants further in-depth investigation.
- (5) In our experiments, we used the default red agent in the CybORG environment. Exploring joint or adversarial training between blue and red agents is worth further investigation in future work.

It should be noted that, compared with current state-of-the-art reinforcement learning methods, our proposed HC-MARL framework achieves significant improvements. However, when compared with generative approaches such as large language models, our method suffers from insufficient interpretability in decision-making, although LLM-based methods typically exhibit lower decision efficiency and rewards.

Author Contributions: Conceptualization, Z.Q. and J.H.; methodology, Z.Q. and J.H.; software, Z.Q.; validation, Z.Q., Z.L. and T.X.; formal analysis, B.W.; investigation, Z.Q.; resources, Z.Q.; data curation, Z.L. and T.X.; writing—original draft preparation, Z.Q.; writing—review and editing, Z.Q. and J.H.; visualization, Z.Q. and Z.L.; supervision, J.H.; project administration, B.W.; funding acquisition, J.H. and B.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The code presented in this study is available on request from the corresponding author.

Acknowledgments: During the preparation of this manuscript, the authors used DeepSeek-V4 for linguistic polishing. The authors have reviewed and edited the output and take full responsibility for the content of this publication.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Kiely, M.; Ahiskali, M.; Borde, E.; Bowman, B.; Bowman, D.; Van Bruggen, D.; Cowan, K.; Dasgupta, P.; Devendorf, E.; Edwards, B.; et al. Exploring the efficacy of multi-agent reinforcement learning for autonomous cyber defence: A CAGE challenge 4 perspective. In *Proceedings of the AAAI Conference on Artificial Intelligence*; AAAI: Washington, DC, USA, 2025; Volume 39, pp. 28907–28913. [\[CrossRef\]](#)
2. Kiely, M.; Bowman, D.; Standen, M.; Moir, C. On Autonomous Agents in a Cyber Defence Environment. *arXiv* **2023**, arXiv:2309.07388. [\[CrossRef\]](#)
3. Kunz, T.; Fisher, C.; Novara-Gsell, J.L.; Nguyen, C.; Li, L. A Multiagent CyberBattleSim for RL Cyber Operation Agents. *arXiv* **2023**, arXiv:2304.11052. [\[CrossRef\]](#)
4. Landolt, C.R.; Würsch, C.; Meier, R.; Mermoud, A.; Jang-Jaccard, J. Multi-Agent Reinforcement Learning in Cybersecurity: From Fundamentals to Applications. *arXiv* **2025**, arXiv:2505.19837. [\[CrossRef\]](#)
5. Nguyen, T.T.; Reddi, V.J. Deep Reinforcement Learning for Cyber Security. *IEEE Trans. Neural Netw. Learn. Syst.* **2023**, *34*, 3779–3795. [\[CrossRef\]](#) [\[PubMed\]](#)
6. Oesch, S.; Austria, P.; Chaulagain, A.; Weber, B.; Watson, C.; Dixon, M.; Sadovnik, A. The Path to Autonomous Cyber Defense. *arXiv* **2024**, arXiv:2404.10788. [\[CrossRef\]](#)
7. Wang, M.; Dechene, R. Multi-Agent Actor-Critics in Autonomous Cyber Defense. *arXiv* **2024**, arXiv:2410.09134. [\[CrossRef\]](#)

8. Standen, M.; Lucas, M.; Bowman, D.; Richer, T.J.; Kim, J.; Marriott, D. CybORG: A Gym for the Development of Autonomous Cyber Agents. *arXiv* **2021**, arXiv:2108.09118. [\[CrossRef\]](#)
9. Singh, A.V.; Rathbun, E.; Graham, E.; Oakley, L.; Boboila, S.; Chin, P.; Oprea, A. Hierarchical multi-agent reinforcement learning for cyber network defense. In *Reinforcement Learning Conference; IFAAMAS*: Richland, SC, USA, 2025.
10. Tang, Y.; Sun, J.; Wang, H.; Deng, J.; Tong, L.; Xu, W. A method of network attack-defense game and collaborative defense decision-making based on hierarchical multi-agent reinforcement learning. *Comput. Secur.* **2024**, *142*, 103871. [\[CrossRef\]](#)
11. Hürten, T.; Loevenich, J.F.; Spelter, F.; Adler, E.; Braun, J.; Moxon, L.; Gourellet, Y.; Lefeuvre, T.; Lopes, R.R.F. Hierarchical multi-agent reinforcement learning for autonomous cyber defense in coalition networks. In *MILCOM 2024—2024 IEEE Military Communications Conference (MILCOM)*, Washington, DC, USA; IEEE: Piscataway, NJ, USA, 2024; pp. 176–181. [\[CrossRef\]](#)
12. Alshamrani, A. Federated hierarchical MARL for zero-shot cyber defense. *PLoS ONE* **2025**, *20*, e0329969. [\[CrossRef\]](#) [\[PubMed\]](#)
13. Liu, Z.; Tu, J.; Hong, Y.; Xiong, L.; Jin, Y.; Tang, Y.; Li, F. HCPO: Hierarchical Conductor-Based Policy Optimization in Multi-Agent Reinforcement Learning. *arXiv* **2025**, arXiv:2511.12123. [\[CrossRef\]](#)
14. Castro, S.R.; Campbell, R.; Lau, N.; Villalobos, O.; Duan, J.; Cardenas, A.A. Large Language Models are Autonomous Cyber Defenders. *arXiv* **2025**, arXiv:2505.04843. [\[CrossRef\]](#)
15. Mukherjee, S.; Chatterjee, S.; Purvine, E.; Fujimoto, T.; Emerson, T. Large language model-based reward design for deep reinforcement learning-driven autonomous cyber defense. *arXiv* **2025**, arXiv:2511.16483. [\[CrossRef\]](#)
16. Ahmed, H.; Iqbal, S.; Pinto Neto, E.C.; Buffett, S.; Sultana, M.; Taylor, A. Autonomous multi-agent cyber defense: A novel approach using reinforcement learning with hierarchical LLM critics. In *2025 Cyber Awareness and Research Symposium (CARS)*, Grand Forks, ND, USA; IEEE: Piscataway, NJ, USA, 2025; pp. 1–5. [\[CrossRef\]](#)
17. Paolo, G.; Benechehab, A.; Cherkaoui, H.; Thomas, A.; Kégl, B. TAG: A Decentralized Framework for Multi-Agent Hierarchical Reinforcement Learning. *arXiv* **2025**, arXiv:2502.15425. [\[CrossRef\]](#)
18. Fox, D.; McCollum, C.; Arnoth, E.; Mak, D. *Cyber Wargaming: Framework for Enhancing Cyber Wargaming with Realistic Business Context*; The Homeland Security Systems Engineering and Development Institute (HSEDI)TM: McLean, VA, USA, 2018.
19. He, J.; Lian, X.; Qi, Q.; Zhang, H. Discussion on Key Technologies of Large-Scale Network Confrontation Wargaming System. *Commun. Technol.* **2018**, *51*, 450–456.
20. Haggman, A. *Cyber Wargaming: Finding, Designing, and Playing Wargames for Cyber Security Education*. Ph.D. Thesis, University of London, London, UK, 2019. Available online: <https://pure.royalholloway.ac.uk/en/publications/cyber-wargaming-finding-designing-and-playing-wargames-for-cyber-> (accessed on 27 July 2026).
21. Curry, J.; Drage, N. *The Handbook of Cyber Wargames: Wargaming the 21st Century*; History of Wargaming Project: London, UK, 2020.
22. Chen, S. *An Analysis of Cyber Wargaming: Current Games, Limitations, and Recommendations*. Bachelor's Thesis, Claremont McKenna College, Claremont, CA, USA, 2022.
23. Roche, E. Cyber wargaming: Research and education for security in a dangerous digital world. *J. Strateg. Secur.* **2025**, *18*, 195.
24. Yang, Y.; Wu, J.; Gao, Z.; Liang, Z.; Hong, C.; Li, P.; Zhang, Y. Wargaming Technology Towards Cybersecurity Threat Evolution in Power Information Network. *South. Power Syst. Technol.* **2025**, *19*, 52–62. [\[CrossRef\]](#)
25. Foley, M.; Hicks, C.; Highnam, K.; Mavroudis, V. Autonomous Network Defence using Reinforcement Learning. In *Proceedings of the 2022 ACM on Asia Conference on Computer and Communications Security, Nagasaki, Japan*; ACM: New York, NY, USA, 2022; pp. 1252–1254. [\[CrossRef\]](#)
26. Hannay, J. John-Cardiff/-Cyborg-Cage-2, 2025. Available online: <https://github.com/john-cardiff/-cyborg-cage-2> (accessed on 27 July 2026).
27. Wiebe, J.; Mallah, R.A.; Li, L. Learning Cyber Defence Tactics from Scratch with Multi-Agent Reinforcement Learning. *arXiv* **2023**, arXiv:2310.05939. [\[CrossRef\]](#)
28. Wang, Q.; He, Z.; Shi, H. Simplifying communication control: A cooperative multi-agent reinforcement learning framework based on group decision-making. *J. King Saud. Univ. Comput. Inf. Sci.* **2025**, *37*, 317. [\[CrossRef\]](#)
29. Palmer, G.; Parry, C.; Harrold, D.J.B.; Willis, C. Deep Reinforcement Learning for Autonomous Cyber Defence: A Survey. *arXiv* **2024**, arXiv:2310.07745. [\[CrossRef\]](#)
30. Lazer, S.J.; Aryal, K.; Gupta, M.; Bertino, E. A survey of agentic AI and cybersecurity: Challenges, opportunities and use-case prototypes. *arXiv* **2026**, arXiv:2601.05293. [\[CrossRef\]](#)
31. Oliehoek, F.A.; Amato, C. *A Concise Introduction to Decentralized POMDPs*; Springer International Publishing: Cham, Switzerland, 2016. [\[CrossRef\]](#)

32. TTCP CAGE Working Group. TTCP CAGE Challenge 4. 2023. Available online: <https://github.com/cage-challenge/cage-challenge-4> (accessed on 27 July 2026).
33. Li, Y.; Tang, S.; Chen, R.; Yu, F.X.; Jiang, G.; Imani, M.; Bastian, N.D.; Lan, T. ACDZero: Graph-embedding-based tree search for mastering automated cyber defense. *arXiv* **2026**, arXiv:2601.02196. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.