electronics

Article

Multi-Agent Hierarchical Reinforcement Learning for PTZ Camera Control and Visual Enhancement

Zhonglin Yang [1,2], Huanyu Liu [1,*], Hao Fang [1], Junbao Li [1] and Yutong Jiang [2]

[1] Faculty of Computing, Harbin Institute of Technology, Harbin 150006, China; 22b903105@stu.hit.edu.cn (Z.Y.); 22s103145@stu.hit.edu.cn (H.F.); lijunbao@hit.edu.cn (J.L.)
[2] Information and Control Technology Department, China North Vehicle Research Institute, Beijing 100072, China; jiangyutong@bit.edu.cn
* Correspondence: liuhuanyu@hit.edu.cn

Abstract

Border surveillance, as a critical component of national security, places increasingly stringent demands on the target perception capabilities of video monitoring systems, especially in wide-area and complex environments. To address the limitations of existing systems in low-confidence target detection and multi-camera collaboration, this paper proposes a novel visual enhancement method for cooperative control of multiple PTZ (Pan–Tilt–Zoom) cameras based on hierarchical reinforcement learning. The proposed approach establishes a hierarchical framework composed of a Global Planner Agent (GPA) and multiple Local Executor Agents (LEAs). The GPA is responsible for global target assignment, while the LEAs perform fine-grained visual enhancement operations based on the assigned targets. To effectively model the spatial relationships among multiple targets and the perceptual topology of the cameras, a graph-based joint state space is constructed. Furthermore, a graph neural network is employed to extract high-level features, enabling efficient information sharing and collaborative decision-making among cameras. Experimental results in simulation environments demonstrate the superiority of the proposed method in terms of target coverage and visual enhancement performance. Hardware experiments further validate the feasibility and robustness of the approach in real-world scenarios. This study provides an effective solution for multi-camera cooperative surveillance in complex environments.

Keywords: hierarchical reinforcement learning; PTZ cameras; graph neural networks; visual enhancement; cooperative perception

Academic Editor: Franco Cicirelli

Received: 11 August 2025
Revised: 9 September 2025
Accepted: 16 September 2025
Published: 26 September 2025

Citation: Yang, Z.; Liu, H.; Fang, H.; Li, J.; Jiang, Y. Multi-Agent Hierarchical Reinforcement Learning for PTZ Camera Control and Visual Enhancement. *Electronics* **2025**, *14*, 3825. https://doi.org/10.3390/electronics14193825

1. Introduction

Border security is essential to a nation's territorial integrity and social stability. In the face of threats such as illegal crossings by individuals, vehicles, or goods, the deployment of video surveillance systems has become a crucial technological means for border security protection [1]. However, due to the vast geographical span and complex terrain of borderlines, traditional video surveillance systems still face numerous technical bottlenecks when dealing with illegal intrusions, making it difficult to achieve efficient, comprehensive, and blind-spot-free monitoring.

In recent years, cameras with Pan–Tilt–Zoom (PTZ) capabilities have been widely adopted in the video surveillance field due to their high flexibility [2]. Compared to traditional fixed-angle cameras, PTZ cameras can dynamically adjust their field of view (FOV), enhancing the resolution of moving targets and better adapting to monitoring

demands in complex environments. However, the monitoring capacity of a single PTZ camera is limited—its viewing angle and coverage area are insufficient for large-scale, dynamic scenarios. As a result, coordinated control of multiple PTZ cameras has emerged as a key research direction for enhancing border surveillance capabilities.

Nevertheless, several critical challenges remain in multi-PTZ camera collaborative monitoring. First, while existing surveillance devices often integrate visual perception algorithms based on object detection, they largely rely on manual joystick and keyboard control or on pre-set automatic patrol programs [3]. This leads to a lack of proactive detection and supplementary information for low-confidence targets, causing potential threats to go undetected or unconfirmed in time, thereby undermining overall surveillance performance (as shown in Figure 1a). Second, given the large scale of border areas, multiple threat targets may appear simultaneously within different cameras' fields of view. Without proper allocation of monitoring tasks among the cameras, critical targets may be overlooked, reducing the overall effectiveness of the surveillance system (as shown in Figure 1b).

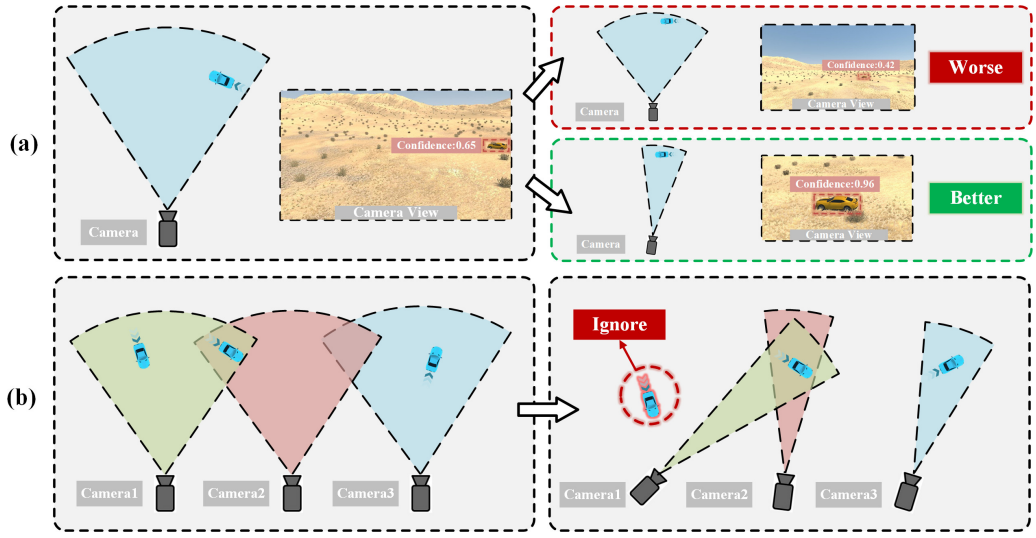

Figure 1. Challenges in multi-PTZ camera collaborative surveillance in border scenarios. (**a**) The impact of lacking active PTZ camera adjustment on surveillance performance, where a comparison is shown between poorer and better camera views for detecting low-confidence targets. (**b**) The problem of target omission in multi-PTZ camera collaborative surveillance, indicating that even with multiple camera fields of view, targets may still be overlooked without effective coordination.

Therefore, how to efficiently coordinate multiple PTZ cameras to complete the collaborative visual enhancement task aimed at improving the detection confidence of low-confidence targets in border scenarios has become the core issue of current research.

To address these challenges, collaborative mechanisms among multiple cameras have become a research hotspot in recent years [4,5]. In particular, multi-agent reinforcement learning (MARL) provides a technical foundation for distributed coordination [6]. However, MARL still faces bottlenecks in large-scale scenarios, such as high-dimensional state spaces, and difficulties in reward allocation [7].

To this end, this paper introduces a hierarchical reinforcement learning (HRL) approach and proposes a two-level hierarchical framework for collaborative visual enhancement using multiple PTZ cameras, as illustrated in Figure 2. This framework leverages high-level agents to guide low-level agents, thereby strengthening the communication and cooperation mechanisms among PTZ cameras. It facilitates information sharing and joint decision-making, which in turn improves overall monitoring coverage and inter-camera collaboration efficiency. This architecture is well-suited to support optimal execution of

visual enhancement tasks in complex border environments, advancing the development of intelligent border security systems.

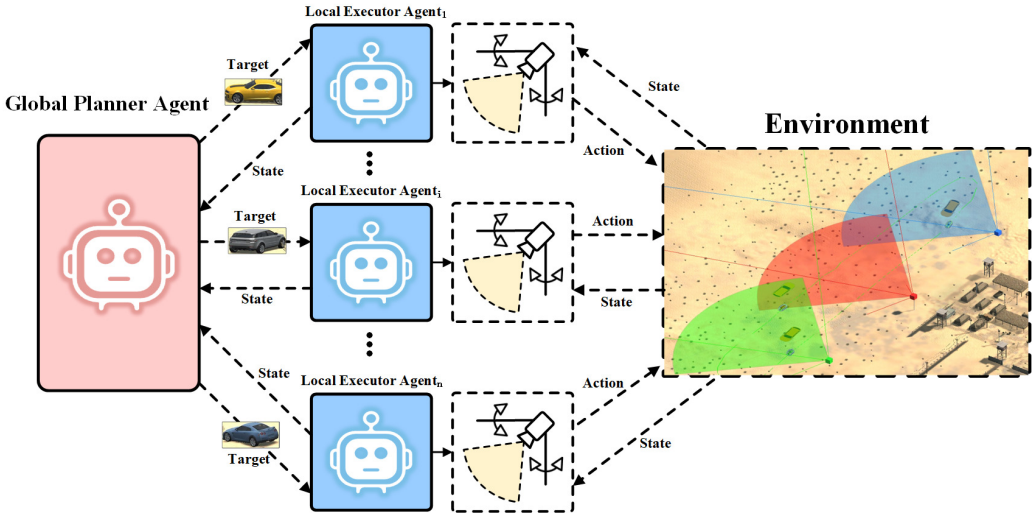

Figure 2. Two-level hierarchical framework for multi-PTZ camera collaborative visual enhancement.

The main contributions of this paper are as follows:

1. We design a two-level hierarchical reinforcement learning architecture consisting of a high-level Global Planner Agent and low-level Local Executor Agents. The high-level agent assigns optimal attention targets to each PTZ camera from a global perspective, while the low-level agents autonomously decide on specific control actions to perform visual enhancement, enabling cross-scale, multi-target collaborative perception to improve task execution efficiency.
2. To effectively capture the topological relationships between cameras and targets, we propose a graph-based joint state space and introduce a graph neural network model to learn complex structural relationships, extract high-level features, and enhance the learning of inter-node dependencies.
3. We develop a simulation environment that mimics real-world scenarios for training and evaluation and implement a corresponding hardware interface to validate the proposed method's transferability and feasibility in real-world applications.

The remainder of this paper is organized as follows. Section 2 reviews related work. Section 3 presents the proposed method in detail. Section 4 discusses the experimental setup, results, and analysis. Section 5 concludes this paper with a summary of contributions and findings.

2. Related Work

2.1. Multi-Agent Reinforcement Learning

MARL has emerged as a significant research direction in artificial intelligence in recent years. It combines reinforcement learning with multi-agent systems, aiming to enable multiple agents to learn in complex environments to achieve either collaborative cooperation or competitive confrontation, thereby reaching their individual or shared goals. With the rapid advancement of AI technologies, MARL has achieved notable success in various application scenarios [8,9]. However, the inherent complexity of MARL has also led to diverse research methodologies and technical challenges.

Current research methods in MARL mainly follow several technical pathways. Independent Reinforcement Learning (IRL), as a foundational paradigm, requires each agent to independently run a single-agent algorithm [10]. Although simple to implement, IRL has

inherent limitations. Because each agent treats others as part of the environment, this disregard for interactions often leads to unstable learning processes and policy conflicts—issues that are especially prominent in cooperative tasks [11]. To improve collaborative performance, value decomposition methods have been proposed, which aim to decompose the global value function into local value functions. For instance, Value Decomposition Networks (VDNs) [12] use a linear summation for efficient computation, while QMIX [13] employs a mixing network for nonlinear decomposition. Both approaches seek to approximate global optima while preserving individual policy autonomy.

Another class of methods emphasizes inter-agent communication. Jiang et al. [14] proposed a Graph-based Proximal Policy Optimization (GPPO) algorithm that introduces graph topology matrices to model the communication relationships among agents. This approach effectively addresses communication coverage issues and enhances inter-agent information exchange, thereby improving policy learning in reinforcement learning. Although these communication-based methods enhance cooperation, they also face real-world constraints such as communication costs and noise interference.

The Centralized Training with Decentralized Execution (CTDE) framework aims to balance training efficiency with execution flexibility. A representative algorithm is MADDPG [15], which incorporates global state information to handle continuous action spaces. Its paradigm of "centralized training, decentralized execution" has become the mainstream architecture. Another notable example is MAPPO [16], which combines the advantages of the PPO algorithm [17] and improves cooperative performance through centralized training and decentralized policy execution.

Compared to single-agent systems, MARL faces key challenges, mainly in terms of environmental dynamics and credit assignment. Due to continuous interactions among agents, the environment becomes exponentially more complex, necessitating effective modeling of the evolving policies of other agents to maintain stability in highly dynamic settings. A more intrinsic difficulty lies in the credit assignment mechanism, how to accurately decompose global rewards down to the individual agent level, which directly affects the learning efficiency and convergence of cooperative systems. As the number of agents increases, these problems evolve into computational bottlenecks: the state space grows explosively, leading to the curse of dimensionality, while the geometric increase in communication overhead severely limits system scalability.

To tackle these challenges, two main technical paths have been proposed. Fully decentralized systems reduce communication overhead via point-to-point messaging, but because agents' policies are interdependent, issues like system instability and policy non-convergence become more prominent. The CTDE framework alleviates some of the instability during training by leveraging global information, yet it does not fundamentally solve the mathematical difficulties of credit assignment. Especially in evaluating individual agent contributions, there remains a lack of a general theoretical framework.

These challenges reflect the dialectical nature of multi-agent systems: ensuring individual autonomy for distributed decision-making while also establishing effective coordination mechanisms to handle environmental complexity.

HRL has emerged as a promising solution to some of these problems [18]. HRL decomposes complex tasks into multiple subtasks, significantly reducing the size of the state and action spaces that each agent must explore. In multi-agent hierarchical structures, high-level policies can provide guidance for low-level policies, reducing the complexity involved in directly assigning global rewards. Furthermore, hierarchical structures simplify complex tasks into more manageable subtasks, mitigating interference during the learning process and enhancing both the system's stability and learning efficiency [19].

Unlike previous studies, this work further combines HRL with graph neural network (GNN) modeling. Although this approach has been applied in some related research [20], here it is specifically optimized for multi-camera target task allocation and collaborative control in border scenarios, with a particular focus on improving visual perception performance for low-confidence targets.

2.2. Reinforcement Learning-Based Autonomous Cooperative Decision-Making for Multiple PTZ Cameras

In recent years, an increasing number of studies have applied reinforcement learning to the cooperative decision-making of multiple PTZ cameras, aiming to optimize their performance in specific tasks. Ci et al. [21] employed a multi-agent reinforcement learning approach to control multiple cameras for collaborative 3D human pose estimation. They introduced a cooperative triangulation contribution reward mechanism to address the credit assignment problem in multi-camera systems. However, their method suffers from sharply increasing computational costs when the number of agents exceeds five, limiting its scalability in large-scale scenarios. Masihullah et al. [22] proposed a decentralized PTZ camera network collaboration strategy based on graph learning, which dynamically learns and leverages graph structures during vehicle tracking, effectively improving the tracking performance of multi-camera systems. Darázs et al. [23] utilized MARL to control a network of PTZ cameras for dynamic intruder detection, behavior tracking, and minimizing sensor usage costs. Through cooperative optimization, they enhanced the overall efficiency of the surveillance system. Hou et al. [24] addressed the problem of camera layout optimization in multi-view pedestrian detection and proposed a Transformer-based configuration generation method. This approach leverages reinforcement learning to automatically explore camera positions and viewpoints, aiming to improve coverage and detection accuracy. Kim et al. [25] proposed a deep reinforcement learning-based cooperative control method for multiple PTZ cameras, in which control values for each camera are generated based on recognition information from an object detection model. Yin et al. [26] introduced a multi-agent reinforcement learning-based system for active multi-camera object tracking (Effi-MAOT). This system employs an intelligent switch and attention mechanism to dynamically select cooperative cameras, significantly reducing bandwidth consumption and improving tracking performance for high-speed targets. Li et al. [27] developed a framework that integrates visual and pose information, applying deep reinforcement learning techniques to enable multiple PTZ cameras to collaboratively perform object tracking tasks. Wang et al. [28] proposed a CTDE-based multi-agent reinforcement learning method for face-tracking tasks using a multi-robot camera array. Their approach employs a Soft Actor–Critic (SAC) model combined with self-attention mechanisms and a cooperative reward design to enhance multi-camera collaboration efficiency and global state modeling capability. Veesam et al. [29] proposed a spatiotemporal framework integrating Multi-Scale Graph Attention Networks with a reinforcement learning-based Dynamic Camera Attention Transformer, which follows the CTDE paradigm to enable adaptive focus reallocation across cameras and significantly improve anomaly detection efficiency in crowded urban scenes. Álvaro et al. [30] proposed a distributed PTZ camera control architecture based on a multi-agent system, allowing each camera to maintain a high degree of autonomy and make final task execution decisions within its own management system. Additionally, their architecture provides a method to coordinate multiple PTZ cameras to achieve better surveillance performance, such as tracking different targets or collaboratively gathering as much information as possible about a single target.

Existing research on autonomous cooperative control of multiple PTZ cameras mostly focuses on optimizing collaboration strategies for specific tasks, with less attention paid to modeling task hierarchies. This paper enhances the global task allocation capability

and perception coordination efficiency of multi-camera systems by introducing an HRL architecture and graph-structured modeling.

3. Multi-Agent Cooperative Algorithm for PTZ Cameras Based on Hierarchical Reinforcement Learning

3.1. Overall Framework

This section proposes an HRL algorithm framework designed for multi-agent collaboration. It adopts an option-based approach to coordinate multiple PTZ cameras in jointly performing visual enhancement tasks for multiple targets. The framework decomposes the overall task into two hierarchical levels: a Global Planner Agent (GPA) responsible for target allocation and multiple Local Executor Agents (LEAs) responsible for performing visual enhancement on the assigned targets.

Within this framework, the GPA acts as a high-level policy that centrally plans tasks for all local agents, while the distributed LEAs serve as low-level policies that independently execute specific perception enhancement tasks. The process operates as follows: At each time step t, the GPA collects the observation results o_i^t from all LEAs (where i denotes the index of the local agent) and assigns a target g_i^t to each LEA based on the current global state. Subsequently, each LEA i independently takes actions to enhance the assigned target based on its own observation and the allocated target g_i^t. Once the assigned tasks are completed, the GPA is reactivated to reassign new targets and dispatch tasks. This process iterates until all targets are processed.

Through this hierarchical design, the complex task is decomposed into two subtasks across different temporal scales, allowing the global planning and local execution layers to be trained independently via single-agent reinforcement learning. This structure not only reduces interference during training but also improves learning stability. For instance, in the early stages of training, separating the GPA and LEAs prevents the LEAs from falling into disordered learning due to receiving invalid goals. Furthermore, the hierarchical design enhances system scalability, reduces communication overhead, and maintains high performance.

At each time step t, the joint state is represented as a graph structure composed of all current targets. Specifically, video frames captured by each PTZ camera are first processed by an object detection model to extract target features, which are then used to construct a graph structure based on predefined rules. Meanwhile, the bounding boxes (BBoxes) of detected targets are used to crop target images, and an appearance feature extraction model generates a sequence of appearance features for all targets, serving as input for the GPA's target allocation.

The GPA takes the graph-structured joint state as input and outputs a probability matrix, where each entry represents the likelihood that a given PTZ camera will select a particular global target. Each PTZ camera then selects the target with the highest probability and extracts the corresponding appearance features, which are used by the respective LEA for subsequent visual enhancement.

During execution, each LEA first uses a feature comparator to locate the assigned target within its field of view. It then controls the PTZ camera to perform a series of actions to complete the visual enhancement. It is important to note that a new joint state is only generated after a subset of PTZ cameras have completed their current enhancement tasks. This mechanism ensures the validity of commands and avoids redundant actions, thereby optimizing resource utilization. The overall algorithmic framework is illustrated in Figure 3.

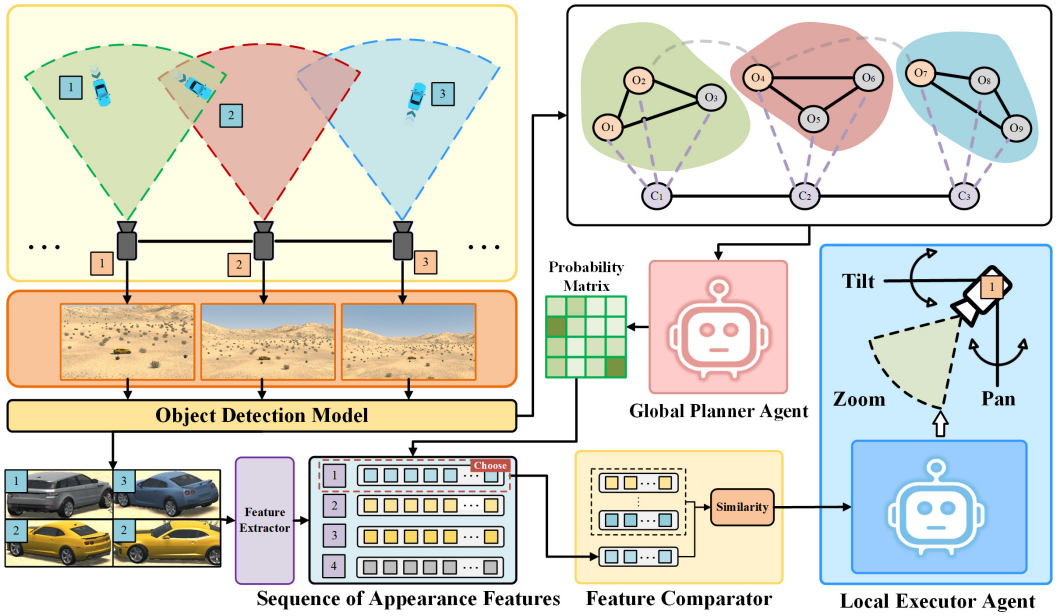

Figure 3. Overall scheme of the hierarchical reinforcement learning-based multi-agent PTZ camera collaboration algorithm. The blue-shaded numbers indicate targets, the orange-shaded numbers indicate cameras, and the purple-shaded numbers indicate feature sequences. Matching blue and purple numbers correspond to a target and its feature sequence.

In hierarchical reinforcement learning, agents at different levels typically operate on distinct temporal scales. To ensure controllability and facilitate system management, this study designs appropriate temporal scales for both the GPA and the LEAs, as illustrated in Figure 4. Specifically, after the GPA completes target assignment, the LEAs sequentially execute their respective visual enhancement tasks. During each action step, the system waits for all LEAs to complete their operations before proceeding to the next decision-making cycle. Although this synchronization mechanism may reduce temporal efficiency to some extent, it guarantees consistency in inter-agent communication and stability in cooperative execution.

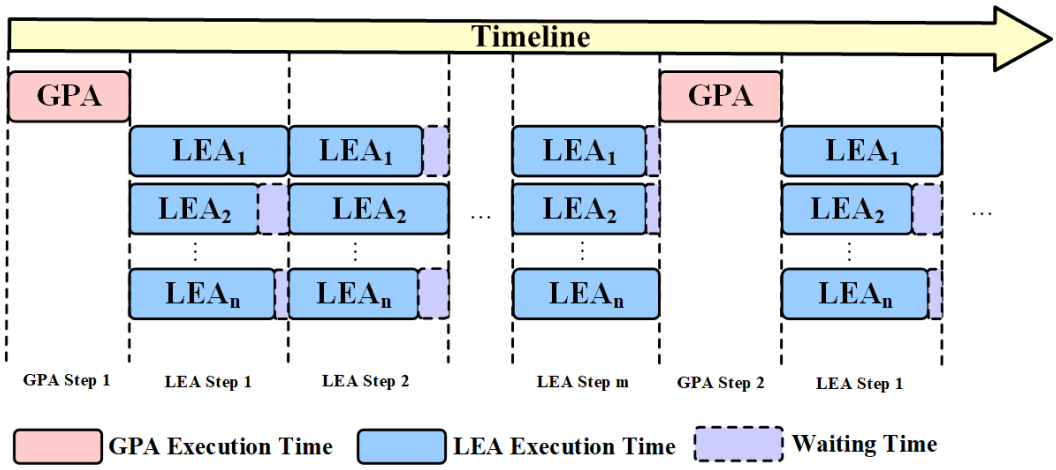

Figure 4. Agent runtime timescale.

Moreover, when a subset of LEAs complete their current visual enhancement tasks, the GPA reallocates targets based on the updated global target distribution. However, if a particular LEA has not yet completed its assigned task, the new allocation will not take effect for that agent, which must continue executing its original task until completion before accepting a new assignment. This mechanism not only ensures continuity in task

execution and optimizes resource utilization but also introduces flexibility for dynamic task reassignment, thereby improving overall system efficiency.

3.2. Local Executor Agent

It is worth noting that the LEA in this study is based on our previous work [31], where the core objective is to control PTZ cameras using reinforcement learning strategies. The LEA performs a series of actions—such as zooming, panning, and tilting—to enhance the detection confidence of low-confidence targets within the field of view. By jointly optimizing the type and duration of PTZ actions, the method significantly improves the agent's decision-making generalizability across various scenarios. The basic framework is shown in Figure 5.

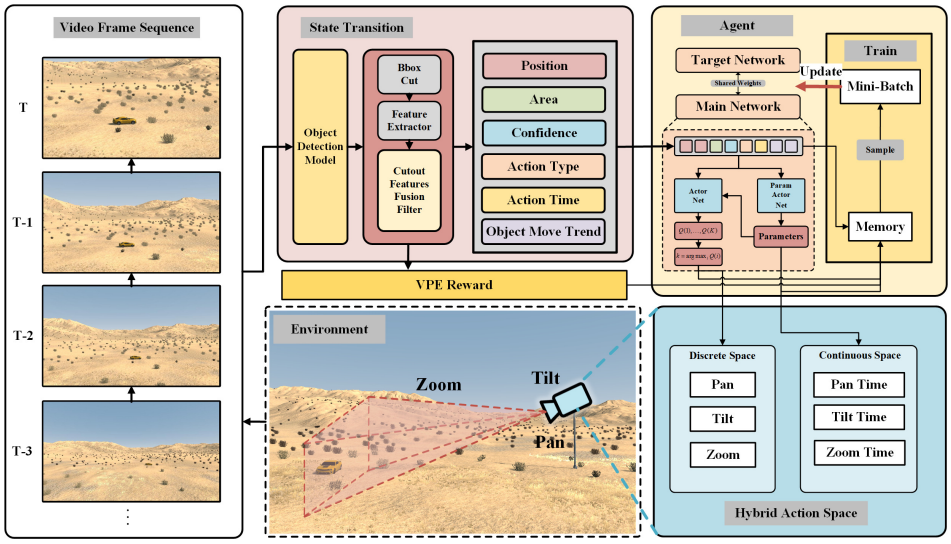

Figure 5. The framework of the Local Executor Agent.

The framework designs the action space as a hybrid hierarchical structure consisting of two levels: the type of action executed by the PTZ camera and its duration. An action is defined as $a = (a_{type}, t_{duration})$, jointly specifying both the action type a_{type} and the execution duration $t_{duration}$, thereby enabling more flexible and fine-grained control strategies.

In terms of state space design, key features such as the target's position, size ratio, and motion trend are selected. The LEA adopts a compact yet informative representation to characterize the target's status, and the overall state vector is defined as

$$S_t = \left[x_{offset},\ y_{offset},\ area_{target},\ c_{target}, a_{t-1},\ d_{t-1},\ x_{trend},\ y_{trend}\right] \tag{1}$$

The state components are computed as follows:

$$x_{offset} = x_t - x_{\text{obs}}, \quad y_{offset} = y_t - y_{\text{obs}} \tag{2}$$

$$area_{target} = \frac{w_t \times h_t}{W_{\text{obs}} \times H_{\text{obs}}} \tag{3}$$

$$x_{trend} = x_t - x_{t-1}, \quad y_{trend} = y_t - y_{t-1} \tag{4}$$

Here, (x_t, y_t) and (w_t, h_t) denote the center coordinates and size of the target bounding box at time t, while $(x_{\text{obs}}, y_{\text{obs}})$ and $(W_{\text{obs}}, H_{\text{obs}})$ represent the image center and frame size of the camera view. c_{target} is the detection confidence, and (a_{t-1}, d_{t-1}) indicate the type and duration of the previous action. This compact formulation provides a discriminative

and noise-robust description of the target's dynamics, thereby improving the reliability of decision-making.

The reward function comprehensively considers detection confidence, positional information, and target size. It is designed to guide the agent in effectively tracking the target while enhancing the detection of low-confidence targets.

However, the LEA method is only applicable to visual enhancement at the single-camera level. When the task is extended to multi-camera systems, the coordination among different cameras becomes ineffective, making global collaborative optimization difficult. Therefore, there is an urgent need to introduce a new mechanism based on the LEA to enable reasonable allocation and coordinated control of global targets across multiple cameras.

3.3. Global Planner Agent

Based on the constructed LEA, the GPA no longer needs to focus on how to control the cameras to achieve better visual enhancement. Leveraging the decoupling advantage of the hierarchical architecture, the core task of the GPA shifts to reasonably assigning global targets, guiding each LEA to enhance the most suitable target. Under this hierarchical design, the GPA can also be modeled using a single-agent reinforcement learning framework. In this work, we adopt DQN (Deep Q-Network) [32] as the implementation method for the GPA. The following sections will detail the design of the GPA's action space, state space, reward function, and network architecture.

3.3.1. Joint Action Space Design

In this task scenario, the action space of the global planning agent is designed to indicate which target each PTZ camera should select from the global target set. Each PTZ camera may detect multiple targets within its field of view, while the global target set $\mathcal{T} = \{t_1, t_2, \ldots, t_T\}$ comprises all targets detected across the views of all PTZ cameras. Correspondingly, the global feature sequence set $\mathcal{F} = \{f_1, f_2, \ldots, f_T\}$ contains the appearance feature sequences for all the targets in the global set, where each feature sequence f_t corresponds one-to-one with a target $t \in \mathcal{T}$. These feature sequences are generated by a feature extraction model based on our previous work [31], which incorporates a cropped feature fusion-based target selection module. Each feature sequence is a 4096-dimensional vector representing the appearance characteristics of a target.

Assuming there are D PTZ cameras in total, the joint action $a = \{a_1, a_2, \ldots, a_D\}$ corresponds to the set of targets chosen by all cameras. For each camera d, the network computes a relevance score $s_{d,t}$ between camera d and each target t, which is then normalized via a Softmax function to yield a probability distribution over candidate targets:

$$p_d(t) = \frac{\exp(s_{d,t})}{\sum_{t' \in \mathcal{T}} \exp(s_{d,t'})}, \quad \forall d \in \{1, \ldots, D\} \tag{5}$$

In practice, each PTZ camera d selects its target by adopting a maximum probability strategy based on this distribution. That is, the selected target index t_d^* is

$$t_d^* = \arg\max_{t \in \mathcal{T}} p_d(t) \tag{6}$$

Once the target t_d^* is determined, PTZ camera d retrieves the corresponding feature sequence $f_{t_d^*}$ from the global feature set $\mathcal{F}$:

$$f_d = F[t_d^*] = f_{t_d^*}, \quad f_{t_d^*} \in \mathbb{R}^n \tag{7}$$

Here, $F\left[t_d^*\right]$ denotes the feature sequence retrieved from the global feature sequence set $\mathcal{F}$ based on the selected target index t_d^*. $f_{t_d^*}$ represents the feature sequence corresponding to the target t_d^* selected by PTZ camera d. Through the above-defined action space and target selection mechanism, the global planning agent can effectively guide the PTZ cameras to perform rational allocation within the global target set.

3.3.2. Joint State Space Design

To effectively represent target features and characterize spatial relationships among targets, this study adopts a graph-based joint state space design. Graph data structures offer topological advantages that enable efficient capture of both local and global relational information, thereby enhancing the agents' perception capabilities in complex environments.

During the construction of the state space, video frames within the field of view of each PTZ camera are first processed by an object detection model to extract features of the detected targets. Based on these features, a graph structure is generated. Considering that the number of visible targets may vary in real-world scenarios, an input dimension alignment strategy is employed: if the number of targets is insufficient, special placeholder values are used to pad the graph nodes. This ensures consistent input dimensions, preventing mismatches during training and inference.

Nodes in the graph are categorized into target nodes and camera nodes, each represented by an attribute vector that encodes its positional features. For target nodes, the feature vector encodes the spatial offset of the detected target relative to the center of the camera's field of view in the form of $[x_{offset}, y_{offset}]$. For camera nodes, the feature vector is derived from the normalized spatial distribution of PTZ cameras in the environment.

To accurately model spatial relationships among targets, between targets and cameras, and between cameras themselves, this study introduces several types of graph edges and corresponding edge weight computation strategies:

- Intra-camera Target Node Connections: A fully connected structure is applied among all target nodes detected by the same PTZ camera. This configuration captures local spatial correlations and reveals spatial distributions and inter-target relationships. Edge weights are computed as the inverse of the Euclidean distance between target coordinates, such that closer targets have higher weights. This design encourages the model to focus on spatially proximate interactions.
- Camera-to-Target Node Connections: Each camera node is connected to the target nodes it detects. The edge weight is determined based on the relative angular position of the target within the camera's field of view. This encoding allows the graph structure to incorporate directional information, enhancing the model's perception of spatial layout between cameras and targets.
- Inter-camera Connections: Fixed-weight edges are established between neighboring PTZ camera nodes to encode adjacency relationships. This connection facilitates the modeling of the camera topology and promotes inter-camera information exchange.
- Cross-camera Target Connections: To improve cooperative perception across PTZ cameras, a special connection strategy is introduced to link targets detected by different cameras. Specifically, the most spatially proximate targets across camera views are connected. This enables cross-regional information sharing, enhancing the system's environmental awareness. Additionally, when a camera fails to detect any targets, these connections allow it to leverage information from neighboring cameras to support detection and decision-making.

Building upon the above edge construction strategies, we further formalize the graph definition and discuss computational complexity. The node set is defined as

$$\mathcal{V} = \{v_c\}_{c=1}^{C} \cup \{v_t\}_{t=1}^{T},$$

where v_c denotes camera nodes and v_t denotes target nodes. The edge set is given by

$$\mathcal{E} = \mathcal{E}_{\text{intra}} \cup \mathcal{E}_{\text{cam-target}} \cup \mathcal{E}_{\text{inter-cam}} \cup \mathcal{E}_{\text{cross-target}},$$

The proposed graph-based joint state space design not only enables efficient feature extraction but also accurately models the spatial relationships between targets and PTZ cameras in multi-target scenarios. This enhances the agent's environmental awareness and improves decision-making efficiency.

3.3.3. Reward Function Design

The reward function in reinforcement learning serves to evaluate the quality of an agent's action given a particular state. In this task scenario, the reward function is designed to simultaneously promote sufficient target coverage and optimize the visual enhancement effect. Specifically, the reward function consists of two components.

The first component is the target coverage reward, defined based on the coverage rate of targets. Target coverage is calculated as the ratio of the number of targets covered by PTZ cameras to the total number of targets. The formula is as follows:

$$r_t^{coverage} = \frac{1}{m} \sum_{j=1}^{m} I_{j,t} \tag{8}$$

Here, $I_{j,t}$ denotes the coverage status of target j at time step t, where 1 indicates that the target is covered and 0 indicates it is not. m represents the number of targets present in the current scene.

The second component is the visual enhancement reward, defined based on the average confidence scores of the targets after LEAs perform visual enhancement on the targets assigned by the GPA. The formula is as follows:

$$r_t^{confidence} = \frac{1}{n} \sum_{i=1}^{n} C_i \tag{9}$$

In this expression, n denotes the number of PTZ cameras in the scene and C_i represents the confidence score of the detected target for PTZ camera i at the final moment when a portion of PTZ cameras complete their assigned visual enhancement tasks.

Finally, these two components are combined via weighted fusion to produce the final reward, as given by

$$r_t = \lambda r_t^{coverage} + (1-\lambda) r_t^{confidence} \tag{10}$$

where λ is the weighting coefficient, which is set to 0.5 in this study.

This design embodies the integration of global and local objectives: the target coverage reward encourages agents to maximize the spatial coverage of potential threat targets, while the visual enhancement reward ensures that the coverage actions result in meaningful perceptual gains. This prevents situations where targets are "covered but ineffectively enhanced," thereby balancing quantity with quality.

Moreover, the reward function incorporates the confidence feedback from the local execution agents, enabling the global planning agent to adapt its strategy based on local execution outcomes. This feedback mechanism facilitates effective cooperation between global

planning and local execution, which supports better agent collaboration and successful task completion.

3.3.4. Agent Network Design

The agent network designed in this study integrates GNNs, aiming to enable each PTZ camera to dynamically select the optimal target of interest from the global target set. The overall structure is illustrated in Figure 6.

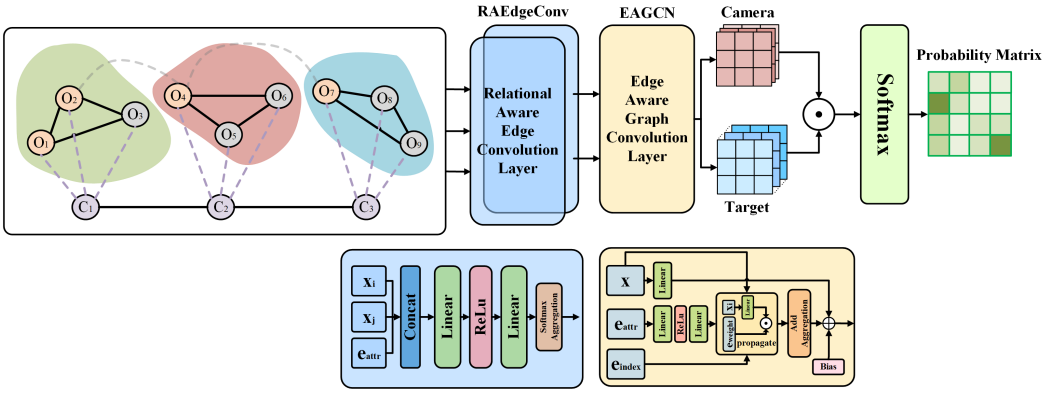

Figure 6. Agent network architecture.

The input graph data, after preprocessing, serves as the representation of the state space and is first passed through two layers of customized Relational-Aware Edge Convolution (RAEdgeConv) to capture local interaction relationships between nodes. Compared with traditional Edge Convolution (EdgeConv) [33], RAEdgeConv enhances the modeling of edge features. While standard EdgeConv mainly treats edges as dynamic local connections to flexibly model the adjacency variation of node features, RAEdgeConv jointly maps the source node features, neighboring node features, and edge attributes using a Multi-Layer Perceptron (MLP). This allows it to more precisely capture the high-order interactive features between PTZ cameras and targets and extract finer-grained details within node relationships.

When aggregating information from neighboring nodes, this module adopts a Softmax-based normalized aggregation strategy. Unlike traditional Max Aggregation or Sum Aggregation, the Softmax strategy provides a finer-grained depiction of the influence weight each neighbor has on the central node. The Softmax normalization dynamically assigns contribution weights from neighboring nodes to the update of the central node, thereby enhancing the flexibility and interpretability of message passing. The two-layer RAEdgeConv stack strengthens the network's capacity to extract expressive features, laying a solid foundation for subsequent global feature fusion.

After capturing local features, the network further employs a customized layer of an Edge-Aware Graph Convolution Network (EAGCN). This module is built upon traditional Graph Convolutional Networks (GCNs) [34] but introduces a dynamic edge weight learning mechanism. Whereas conventional GCNs aggregate node features using fixed weights, the EAGCN uses an MLP to generate edge weights, allowing edge features to modulate the message-passing process. This design not only improves modeling of interactions between targets but also incorporates the relationships between PTZ cameras and targets, yielding richer global graph representations. The inclusion of the EAGCN enables the network to aggregate global structural information more effectively, compensating for the locality-focused nature of RAEdgeConv.

In the final stage of the network, the node features are split into PTZ camera nodes and target nodes. To determine each PTZ camera's selection probability over the global targets,

the network computes the inner product between the feature vectors of each PTZ camera node and each target node, resulting in a relevance score for each camera–target pair. These scores are then normalized using a Softmax function to produce a probability distribution over target selections for each PTZ camera. This mechanism allows the network to dynamically adjust each PTZ camera's target selection strategy based on global information, thus adapting to varying decision-making demands in different environments.

For the GNN modules, both the EAGCN and RAEdgeConv follow a weighted message-passing scheme:

$$\mathbf{h}_v^{(k+1)} = \phi\left(\mathbf{h}_v^{(k)}, \bigoplus_{u \in \mathcal{N}(v)} \psi(\mathbf{h}_u^{(k)}, e_{uv})\right),$$

where $\bigoplus$ is a symmetric aggregation operator. Since the aggregation is independent of the ordering of neighbors, the model is inherently permutation-invariant.

The computational complexity of the proposed EAGCN/RAEdgeConv stack is primarily determined by the number of nodes N and the number of edges E in the constructed graph. For intra-camera connections, a fully connected structure among n_i targets within camera i introduces $O(n_i^2)$ edges, whereas camera-to-target connections contribute $O(n_i)$ edges per camera. Inter-camera and cross-camera connections introduce a fixed and sparse number of additional edges. Overall, the total edge count E scales approximately as $O(\sum_i n_i^2)$.

In the EAGCN layers, edge weights are dynamically learned for each edge, which incurs an additional $O(E \cdot d)$ cost for computing these weights on top of the standard RAEdgeConv message passing. Therefore, the per-layer computational complexity of the combined EAGCN/RAEdgeConv stack is approximately $O(E \cdot d^2 + E \cdot d)$, where d denotes the node feature dimension. Here, the first term accounts for feature aggregation, while the second term corresponds to the computation of dynamic edge weights.

Given the typically moderate number of visible targets per camera in practical scenarios, this graph construction and message passing remain computationally feasible for online inference.

4. Experiments

4.1. Simulation Environment

Training reinforcement learning algorithms typically require frequent interactions with the environment. However, executing this process on physical hardware often incurs high time costs and equipment wear, and it is also difficult to construct ideal training scenarios. To address these challenges, we developed a complete simulation environment based on the Unity3D (2021.3.21f1) engine, which realistically simulates the control process of PTZ cameras and their collaborative mechanisms, as shown in Figure 7.

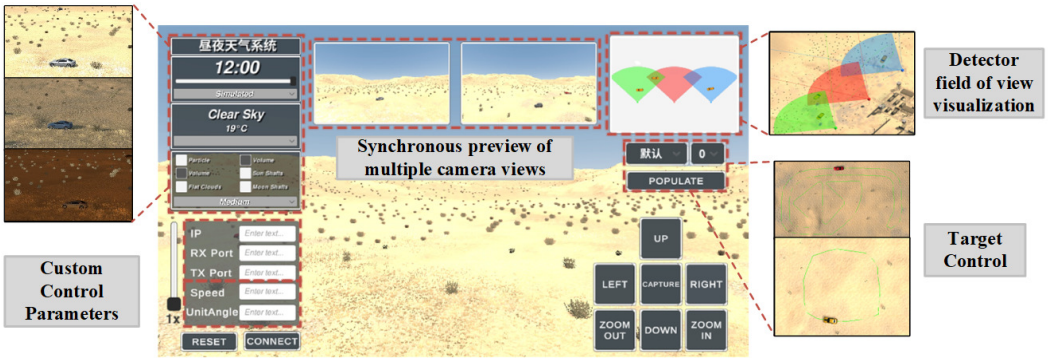

Figure 7. Simulation environment.

This simulation environment supports real-time preview of each camera's field of view and its visual effects. It also allows users to customize various parameters, including PTZ camera control settings, target motion trajectories, and environmental weather conditions. Moreover, the simulation environment is compatible with our re-implemented Gym interface, enabling the transmission of video frames, action commands, and other necessary data to support a full reinforcement learning training pipeline.

4.2. Training Settings and Results

As previously mentioned, the model mainly consists of two layers of RAEdgeConv modules and one layer of an EdgeConv module. During training, Smooth L1 loss is used as the regression loss to optimize the attention distribution between camera nodes and target nodes, providing robustness to outliers and improving training stability. Due to communication constraints in the simulation environment, training the model on an NVIDIA GeForce RTX 3060 GPU (Nvidia Corporation, Santa Clara, CA, USA) takes approximately 12 h, with an average runtime per episode of approximately 43.2 s. Since the LEA directly adopts results from previous studies, it is used as is during the GPA training phase, without requiring additional training.

4.2.1. Training Objective and Loss Function

The training of the GPA follows the DQN framework. The joint action $a = \{a_1, a_2, \ldots, a_D\}$ represents the assignment of targets to all PTZ cameras, where each $a_d = t_d^*$ is selected according to the maximum probability strategy from the camera-specific relevance scores.

The Q-network $Q_\theta(s, a)$ estimates the expected return of taking joint action a in state s. In this implementation, the Q-values of individual cameras are first gathered for the executed actions and then averaged across cameras:

$$Q_\theta(s, a) = \frac{1}{D} \sum_{d=1}^{D} Q_\theta^d(s, a_d), \tag{11}$$

where $Q_\theta^d(s, a_d)$ is the Q-value for camera d choosing action a_d and D is the total number of cameras.

For a transition (s, a, r, s', d) sampled from the replay buffer, the Bellman target is defined as follows:

$$y = r + (1 - done)\, \gamma \max_{a'} Q_{\theta^-}(s', a') \quad \text{with} \quad Q_{\theta^-}(s', a') = \frac{1}{D} \sum_{d=1}^{D} \max_{a'_d} Q_{\theta^-}^d(s', a'_d), \tag{12}$$

where $\gamma \in (0, 1)$ is the discount factor and θ^- denotes the parameters of the target Q-network.

The temporal difference (TD) error is

$$\delta = Q_\theta(s, a) - y, \tag{13}$$

and the Smooth L1 (Huber) loss is applied:

$$\mathcal{L}(\theta) = \mathbb{E}\Big[\ell(\delta)\Big], \quad \ell(\delta) = \begin{cases} \frac{1}{2}\delta^2, & \text{if } |\delta| < 1, \\ |\delta| - \frac{1}{2}, & \text{otherwise.} \end{cases} \tag{14}$$

This loss function is often considered suitable for TD error optimization in DQN training, as it combines the advantages of both L1 and L2 losses: it behaves like an L2 loss for small TD errors, ensuring smooth gradients for stable learning, while behaving like an

L1 loss for large TD errors, which helps reduce the influence of outliers or occasional large Q-value deviations. This property can enhance training stability, particularly in scenarios with varying target detection confidence and multiple interacting cameras.

4.2.2. Episode Initialization and Termination

At the beginning of training, the model's network parameters are initialized and copied to their respective target networks. The agent updates the network parameters every five steps. Targets are randomly generated within the map. Each episode terminates under either of the following conditions:

- The average confidence of all observed targets reaches 0.8 or above for two consecutive steps;
- The total number of steps taken by the agent reaches or exceeds 10.

The detailed hyperparameter settings are shown in Table 1.

Table 1. Hyperparameter design.

Hyperparameter Name	Value
Learning Rate	0.001
Discount Factor	0.90
Replay Buffer Capacity	5000
Batch Size	256
Number of Iterations	1000
Optimizer	Adam
Target Network Update Frequency	5
ϵ-Greedy Strategy	linear decay to 0.1 over first 150 episodes

During the training process over 1000 episodes, the average reward curves under three random seeds for the proposed method are shown in Figure 8. As illustrated, the reward curves eventually converge.

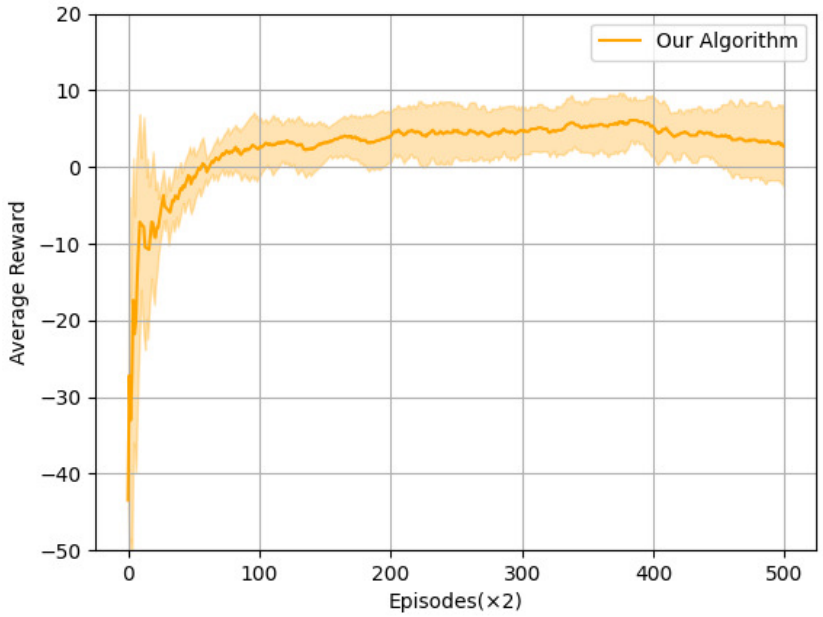

Figure 8. Average training reward.

4.3. Evaluation Metric

To evaluate how well the GPA guides the LEAs to control PTZ cameras in covering global targets, we introduce the Average Cover Rate (ACR). This metric reflects the average of the per-step coverage rates across all episodes. It is defined as follows:

$$ACR = \frac{1}{N}\sum_{i=1}^{N}\bar{C}_i = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{1}{T_i}\sum_{t=1}^{T}\bar{C}_{i,t}\right) \tag{15}$$

where N is the number of experiments, T_i is the number of steps in the i-th experiment, and $R_{i,t}$ is the coverage rate at step t in the i-th experiment.

To assess how well the GPA and LEAs collaborate to complete the visual enhancement tasks on all global targets, we introduce the Visual Enhancement Ratio Achieved (VERA). This metric indicates the proportion of all targets in the experiment for which visual enhancement has been successfully completed (i.e., the target detection confidence reaches 0.9 or above). It is defined as follows:

$$VERA = \frac{N_{enhancement}}{N_{target}} \tag{16}$$

where $N_{enhancement}$ is the number of targets that have completed visual enhancement and N_{target} is the total number of targets in the experiment.

To evaluate the effectiveness of the proposed method in collaboratively enhancing the confidence of low-confidence targets, we introduce Initial Average Confidence (IAC) and Enhanced Average Confidence (EAC). IAC represents the mean and standard deviation of the initial detection confidence values across all targets and experiments, while EAC represents the same statistics after visual enhancement is completed. The definitions are as follows:

$$IAC = \mu_{initial} \pm \sigma_{initial} \tag{17}$$

$$EAC = \mu_{enhance} \pm \sigma_{enhance} \tag{18}$$

Here,

$$\mu_{initial} = \frac{1}{N}\sum_{i=1}^{N} C_i^{initial},$$

$$\sigma_{initial} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(C_i^{initial} - \mu_{initial}\right)^2},$$

$$C_i^{initial} = \frac{1}{M_i}\sum_{j=1}^{M_i} C_{i,j}^{initial}$$

$$\mu_{enhance} = \frac{1}{N}\sum_{i=1}^{N} C_i^{enhance},$$

$$\sigma_{enhance} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(C_i^{enhance} - \mu_{enhance}\right)^2},$$

$$C_i^{enhance} = \frac{1}{M_i}\sum_{j=1}^{M_i} C_{i,j}^{enhance}$$

where N is the number of experiments, M_i is the number of detected targets in the i-th experiment, $C_i^{initial}$ is the mean of the initial confidence values for all targets in the i-th experiment, and $C_i^{enhance}$ is the mean confidence after enhancement in the same experiment.

To assess the efficiency of agent actions, we introduce the Average Step (AS) metric, which reflects the average number of steps taken by the GPA across all experiments. It is defined as follows:

$$AS = \frac{\sum_{i=1}^{N} B_i}{N} \tag{19}$$

where B_i is the number of steps taken in the i-th experiment and N is the total number of experiments.

4.4. Comparative Experiment

To evaluate the effectiveness and rationality of the GPA in global target allocation, as well as the actual performance of the LEA in executing visual enhancement tasks after allocation, we conducted tests in 150 randomly generated experimental scenarios with two to four targets, with target positions randomly distributed within the scene. The target detection algorithm used is YOLOX [35], implemented based on the MMDetection open-source framework [36]. The comparative algorithm was MADDPG [15]. The experimental results are shown in Table 2.

Table 2. Comparative experiment results.

Target Quantity	Method	ACR	VERA	IAC	EAC
2	MADDPG	91.15%	16.67%	0.86 ± 0.01	0.45 ± 0.15
	Ours	**99.52%**	**95.83%**	0.86 ± 0.02	**0.90 ± 0.02**
3	MADDPG	80.96%	30.58%	0.86 ± 0.03	0.54 ± 0.12
	Ours	**95.80%**	**90.21%**	0.85 ± 0.04	**0.91 ± 0.03**
4	MADDPG	85.33%	31.03%	0.87 ± 0.01	0.69 ± 0.14
	Ours	**92.12%**	**74.14%**	0.86 ± 0.04	**0.92 ± 0.00**

Note: Bold values indicate the best performance.

From the results, it can be observed that even with only three PTZ cameras, the proposed algorithm achieves a good ACR across different numbers of targets, indicating that most targets remain within the cameras' field of view most of the time. Notably, when the number of targets is two or three, the VERA reaches a high level, exceeding 90% in both cases. When there are four targets, VERA drops to 74.14%, which is still close to the theoretical maximum of 75%, since with only three PTZ cameras, at most three targets can be simultaneously enhanced. Overall, the results demonstrate that the proposed algorithm efficiently allocates PTZ camera resources and exhibits superior performance.

In contrast, the MADDPG algorithm performs relatively poorly in this task scenario. The experiments show that MADDPG struggles to achieve satisfactory results when performing collaborative visual enhancement for three targets. This is mainly due to the high complexity of the decision space, as agents must simultaneously decide which target the camera should focus on, the type of action to execute, and the duration of that action under the current task setup, making it difficult for the model to learn a stable policy.

This phenomenon further validates the rationality of adopting a hierarchical architecture for complex tasks. By decomposing the overall task into smaller subtasks, the hierarchical method effectively reduces the learning difficulty of each subtask and lessens the complexity of directly assigning global rewards. This facilitates stable convergence of agent policies and improves overall performance.

To provide a more intuitive demonstration of the practical effectiveness of our proposed algorithm, we selected image sequences from several representative scenarios for illustration. First, a scenario with three targets distributed within the scene is shown in Figure 9. It can be observed that the GPA evenly allocates the three targets to different PTZ cameras, with each camera prioritizing the target within its field of view—a strategy consistent with intuitive expectations. As the LEA performs visual enhancement on the assigned targets step by step, the detection confidence gradually increases, eventually exceeding 0.9, validating the algorithm's rationality and effectiveness.

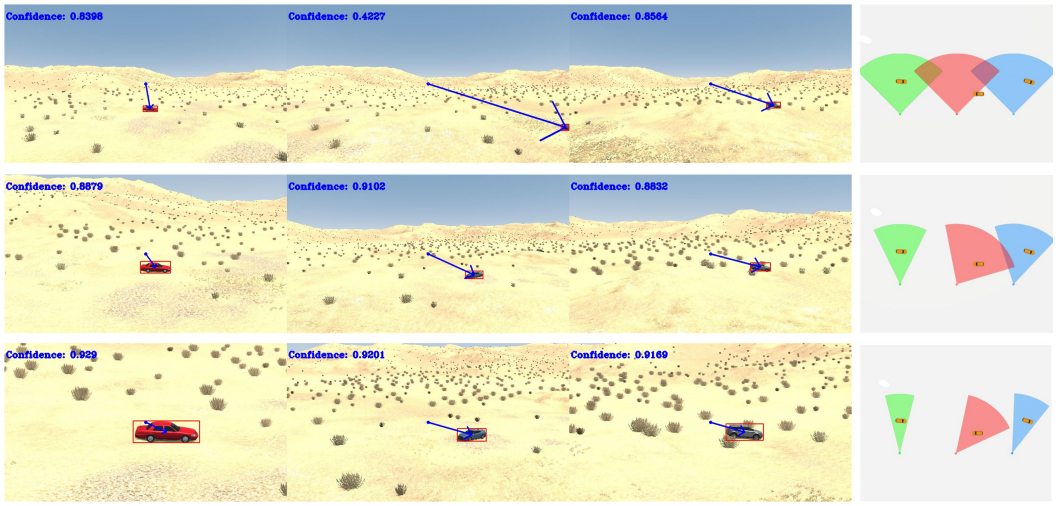

Figure 9. Experimental image sequence for three-target scenarios. The arrows in the images represent the target spatial offset within the joint state space, indicating the displacement of the Bbox centers relative to the center of the camera's field of view.

Next, we present a special case where three targets are present, but two of them fall within the field of view of the same PTZ camera, as shown in Figure 10. In this case, the GPA demonstrates high flexibility and intelligence. It assigns the PTZ camera that initially has no targets in view to focus on one of the two overlapping targets, while the original PTZ camera focuses on the other. As a result, all targets are successfully enhanced by the PTZ cameras, with their confidence levels reaching above 0.9. This strategy maximizes resource utilization and avoids missing any targets, thoroughly validating the algorithm's design and the efficiency of its allocation strategy.

Then, we show a scenario with two targets, as in Figure 11. In this case, some PTZ cameras have no targets within their view, which theoretically would leave them idle. However, the GPA does not leave these cameras idle. Instead, it assigns them to jointly observe the same target as their neighboring PTZ cameras. This strategy is of great practical significance: on one hand, it avoids wasting resources; on the other, it enhances system fault tolerance. If one PTZ camera fails or loses communication, the other can continue enhancing the target, ensuring the robustness and reliability of the system.

Lastly, we show a scenario with four targets, as seen in Figure 12. Since the number of targets exceeds the number of PTZ cameras, one target is theoretically unassignable and cannot be directly enhanced. However, during execution, the unassigned target is still placed within the field of view of a PTZ camera as much as possible.

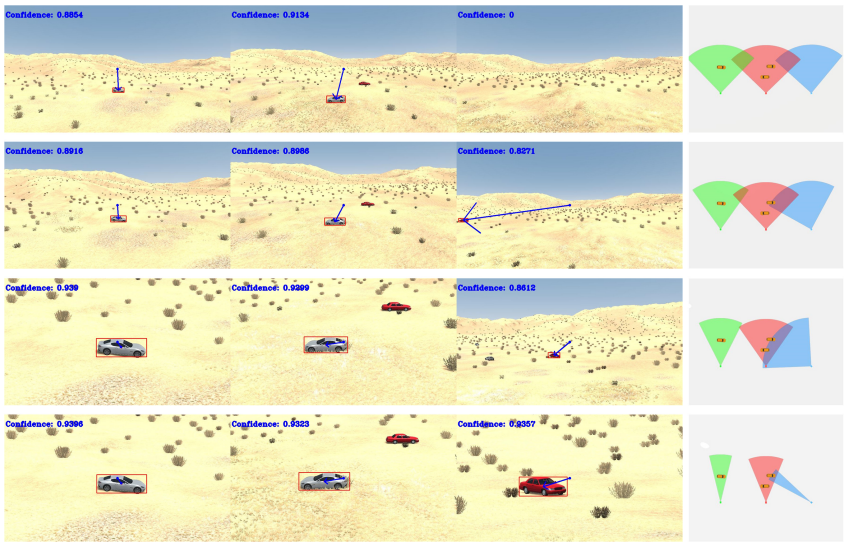

Figure 10. Experimental image sequence for special three-target scenarios. The arrows in the images represent the target spatial offset within the joint state space, indicating the displacement of the Bbox centers relative to the center of the camera's field of view.

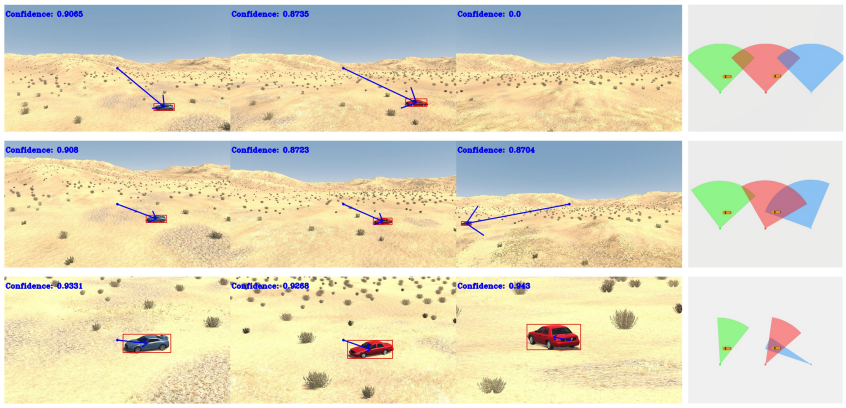

Figure 11. Experimental image sequence for two-target scenarios. The arrows in the images represent the target spatial offset within the joint state space, indicating the displacement of the Bbox centers relative to the center of the camera's field of view.

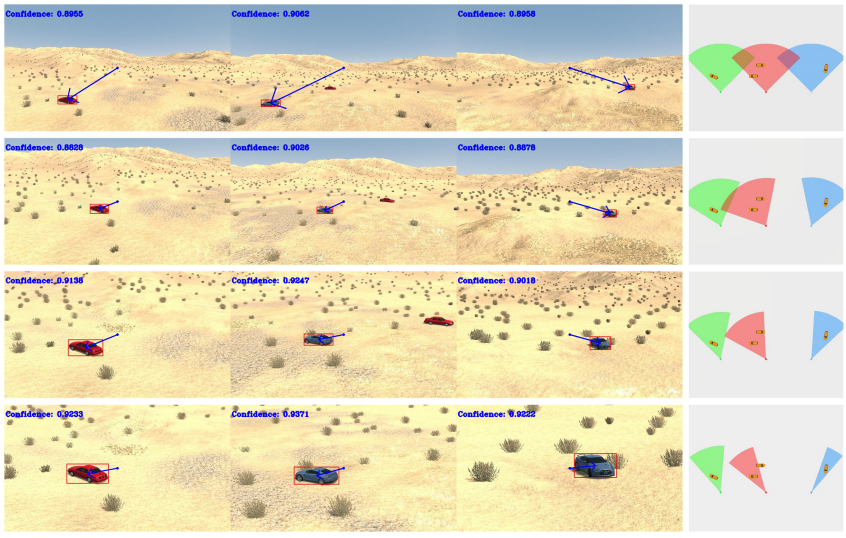

Figure 12. Experimental image sequence for four-target scenarios. The arrows in the images represent the target spatial offset within the joint state space, indicating the displacement of the Bbox centers relative to the center of the camera's field of view.

4.5. Ablation Experiment

4.5.1. Ablation Study on Network Architecture

To verify the effectiveness of the proposed GNN architecture, an ablation experiment on the network structure was designed. A control model without graph structural information—a fully connected neural network (Multi-Layer Perceptron, MLP)—was constructed to evaluate the specific contribution of graph modeling to task performance.

Unlike the previously described model, which performs state modeling and information interaction based on graph structures, the MLP model directly flattens all observed states into a vector for input. In the output part, the MLP network is required to accomplish the same task as the GNN: selecting a target from the global target pool for each camera; i.e., each camera outputs a probability distribution over all global targets, reflecting its focus tendency on targets under the current state.

To ensure fairness, the experiment maintained consistency with the original GNN model in terms of loss function, training strategy, and other aspects, only replacing the network architecture. Experiments were conducted under the same 150 experimental scenarios as with the GNN model to fairly assess the contribution of graph structure modeling to the final performance. The experimental results are shown in Table 3.

Table 3. Network architecture ablation experimental results.

Target Quantity	Architecture	ACR	VERA	IAC	EAC	AS
2	MLP	95.05%	79.17%	0.85 ± 0.04	0.80 ± 0.18	6
	GNN	**99.52%**	**95.83%**	0.86 ± 0.02	**0.90 ± 0.02**	**2**
3	MLP	93.39%	70.64%	0.86 ± 0.03	0.82 ± 0.13	6
	GNN	**95.80%**	**90.21%**	0.85 ± 0.04	**0.91 ± 0.03**	**3**
4	MLP	**92.81%**	65.52%	0.87 ± 0.03	0.90 ± 0.06	5
	GNN	92.12%	**74.14%**	0.86 ± 0.04	**0.92 ± 0.00**	**2**

Note: Bold values indicate the best performance.

The results indicate that when the network structure is replaced by the MLP model, the effectiveness of target allocation deteriorates. This is mainly because the MLP model, in terms of state space design, cannot explicitly model spatial structural information such as relative positional relationships between targets and the connections between cameras and targets. Structurally, the MLP model also fails to extract effective high-order feature information. This validates the effectiveness and rationality of introducing the GNN model and further demonstrates its advantages in multi-camera tasks.

4.5.2. Ablation Study on Reward

To validate the effectiveness of the reward function proposed in this chapter, we designed and conducted an ablation experiment on the reward function. In this experiment, we ablated the part of the reward function related to the LEA's confidence feedback, retaining only the component related to target coverage. The experimental results are shown in Table 4.

From the results, it can be seen that although a reward function based solely on coverage can achieve a relatively good ACR to some extent, the VERA performs poorly, indicating an inability to effectively accomplish the visual enhancement task. This is because, under this reward setting, the GPA cannot obtain effective feedback on task completion from the LEA, making it unable to evaluate the rationality of target allocation. More importantly, relying only on coverage-based rewards fails to provide sufficiently differentiated feedback for the GPA, which hinders its ability to learn appropriate allocation strategies during training—leading to rigid and suboptimal allocation patterns.

Table 4. Reward ablation experimental results.

Target Quantity	$r_t^{coverage}$	$r_t^{confidence}$	ACR	VERA	IAC	EAC	AS
2	✔	✗	84.00%	0%	0.85 ± 0.04	0.00 ± 0.00	10
	✔	✔	**99.52%**	**95.83%**	0.86 ± 0.02	**0.90 ± 0.02**	**2**
3	✔	✗	93.78%	0%	0.86 ± 0.03	0.00 ± 0.00	10
	✔	✔	**95.80%**	**90.21%**	0.85 ± 0.04	**0.91 ± 0.03**	**3**
4	✔	✗	94.75%	0%	0.87 ± 0.02	0.00 ± 0.00	10
	✔	✔	92.12%	**74.14%**	0.86 ± 0.04	**0.92 ± 0.00**	**2**

Note: Bold values indicate the best performance. ✔ indicates the corresponding reward is included; ✗ indicates it is not included.

These experimental results strongly demonstrate the necessity and effectiveness of incorporating the LEA's confidence feedback into the reward function. The addition of the confidence component significantly improves the effectiveness of cooperative visual enhancement, verifying the rationality and feasibility of the reward design.

4.6. Complex Scenario Experiments

To further validate the effectiveness and robustness of the proposed method in more complex environments, occlusions were introduced and targets were assigned certain motion speeds in 100 experimental scenarios. Based on this setup, experiments were conducted to evaluate the algorithm's performance in dynamic and complex situations. The experimental results are presented in Table 5.

Table 5. Complex scenario experimental results.

Target Quantity	ACR	VERA	IAC	EAC
2	100.00%	100.00%	0.83 ± 0.01	0.88 ± 0.03
3	94.38%	88.10%	0.84 ± 0.05	0.91 ± 0.01
4	89.57%	66.96%	0.86 ± 0.03	0.91 ± 0.01

The results show that, except for the case with two targets, the VERA performance in these scenarios noticeably decreases. This decline is mainly due to the large spatial span of the experimental scenes; as targets continue moving, they may enter areas beyond the maximum field of view of the cameras. Additionally, when targets move at higher speeds, some cameras require more action steps to complete the visual enhancement tasks for the targets, which leads to delays in completing enhancement and thus impacts overall performance.

Several representative frames were selected for visualization, as shown in Figure 13. It can be observed that even when targets are occluded and cannot be detected normally, the proposed GPA algorithm can quickly recognize and perform visual enhancement once the targets re-enter the camera's field of view. This robustness is attributed to the GPA's target allocation process, which differentiates targets based on appearance feature vectors extracted by the cropping feature fusion module, thereby endowing the system with a degree of occlusion robustness and re-identification capability.

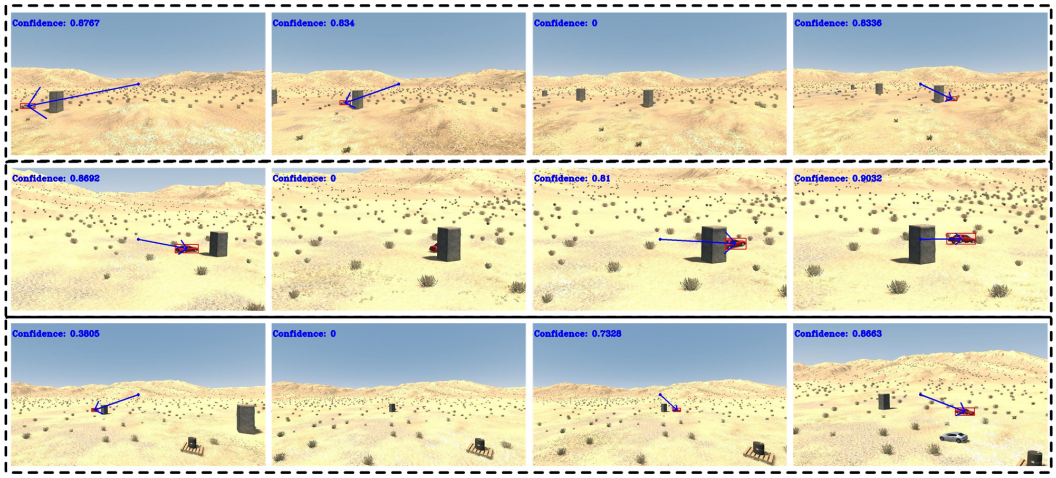

Figure 13. Experimental image sequence for complex scenarios. The arrows in the images represent the target spatial offset within the joint state space, indicating the displacement of the Bbox centers relative to the center of the camera's field of view.

4.7. Hardware Experiment

To verify the feasibility of the proposed approach and promote its practical application, we developed a complete hardware system, as illustrated in Figure 14. The system employs a camera module with 30× optical zoom and integrates a pan–tilt unit supporting the RS485 interface and Pelco-D protocol communication, forming a single hardware device. The hardware specifications are summarized in Tables 6 and 7, with the camera module's model and parameters kept consistent throughout the system. On the software side, we developed a hardware control interface consistent with that of the simulation environment, enabling the transmission of control commands to the real-world hardware and the receipt of feedback on action states. This design minimizes domain shift issues during model deployment and migration.

Table 6. Camera parameters.

Parameter Name	Value
Zoom	30× Optical Zoom
Communication Interface	RS485 Interface
Access Protocol	ONVIF
Operating Temperature	−20 °C 60 °C
Resolution	1920 × 1080

Table 7. Pan–tilt unit parameters.

Parameter Name	Value
Horizontal Rotation Angle	±175°
Vertical Rotation Angle	±35°
Rotation Speed	10°/s
Operating Temperature	−25 °C 50 °C
Communication Method	RS485 Half-Duplex Bus
Protocol	Pelco-D

In actual deployment, the algorithm and models must first be transferred and deployed to a Local Compute Unit (LCU), which is connected to the PTZ camera hardware. This enables the LEA model, running on the LCU, to send control instructions to the PTZ camera. All LCUs communicate with a Global Compute Unit (GCU) via a network switch,

transmitting their local state information. Based on the deployed GPA policy model, the GCU assigns targets to specific PTZ cameras.

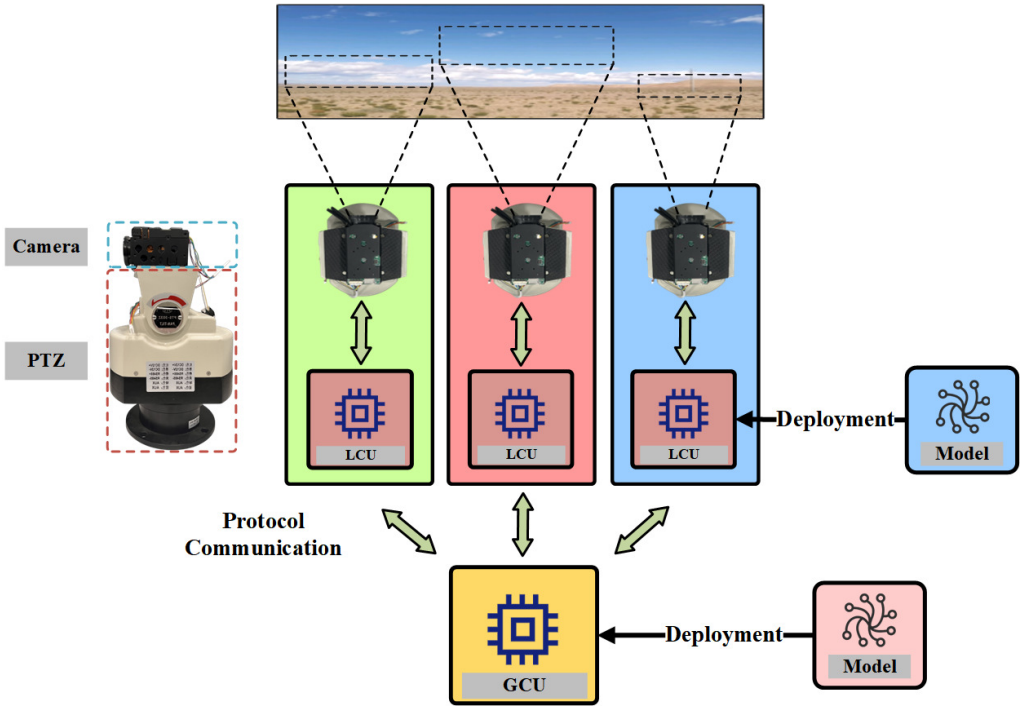

Figure 14. Schematic diagram of the hardware system.

Due to limitations in experimental scenarios and hardware resources, the real-world experiments in this study could not be conducted outdoors in a setting closely matching the simulation environment, nor were outdoor deployment conditions available. Therefore, we selected a relatively simple and controllable indoor scenario, with the primary purpose of verifying the feasibility and adaptability of the algorithm on the hardware system platform. In the experiment, pedestrian targets were used for monitoring, and selected video frames from the execution process are shown in Figure 15.

The experimental results demonstrate that the proposed hierarchical reinforcement learning-based multi-agent PTZ camera coordination algorithm can operate stably and successfully perform collaborative visual enhancement tasks when migrated to the hardware platform. These results confirm the rationality of the algorithm's architecture, the feasibility of the deployment scheme, and the practical application potential of the proposed approach. In future work, we aim to address the challenges of constructing and deploying real-world scenarios to extend the experiments to more complex outdoor environments, further validating the method's practicality and robustness.

It should be noted that, due to the limitations of the experimental scenarios and hardware resources, the hardware validation in this study was primarily conducted on a relatively small-scale multi-PTZ camera network, and the scalability to larger networks has not yet been fully tested. In practical deployments, multiple challenges may still arise. For example, as the number of cameras increases, the system's sensitivity to communication latency in global task allocation and real-time collaboration will significantly increase; hardware heterogeneity in terms of the model, communication protocols, and computing capabilities may lead to inconsistencies in the execution efficiency and accuracy of the algorithm across different devices. In addition, issues such as bandwidth usage, time synchronization, and energy consumption control will also need to be further evaluated and optimized in larger-scale deployments. These challenges will be investigated in future

work in conjunction with a larger-scale hardware platform in order to further verify and enhance the system's practical applicability and robustness.

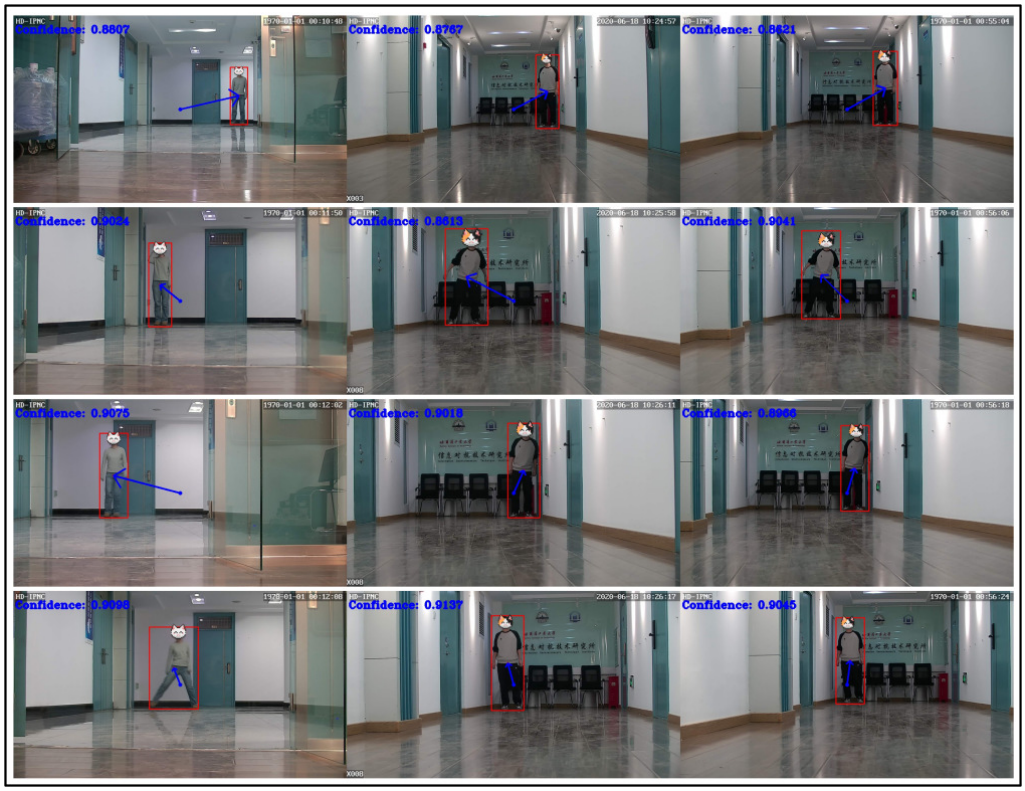

Figure 15. Real-world multi-pedestrian, multi-camera hardware experiment results. The arrows in the images represent the target spatial offset within the joint state space, indicating the displacement of the Bbox centers relative to the center of the camera's field of view.

5. Conclusions

This paper addresses the problem of collaborative perception and visual enhancement of multiple PTZ cameras in border scenarios by proposing a hierarchical reinforcement learning–based multi-agent control framework. By constructing a two-level hierarchical structure comprising a GPA and LEAs, the framework achieves reasonable global target allocation and effective local execution cooperation. Moreover, to enhance the agents' ability to model complex spatial relationships in the environment, a joint state space based on graph structures is designed, and graph neural networks are introduced to extract high-order feature correlations between multiple cameras and multiple targets.

Meanwhile, a composite reward function combining target coverage and visual enhancement effects is proposed to promote effective cooperation between global and local agents. In simulation experiments, the proposed method outperforms traditional multi-agent reinforcement learning approaches in both target coverage and visual enhancement performance, validating the advantages of the hierarchical architecture in task decomposition and reward allocation. Ablation studies further confirm the effectiveness of graph neural networks in state modeling and the necessity of the confidence feedback mechanism in the reward function. Experiments in complex scenarios increase environmental diversity and verify the robustness of the method under conditions such as occlusion and dynamic target changes. The final hardware experiments demonstrate that the method possesses strong practical transferability and application potential.

Future research will consider extending to larger-scale PTZ camera networks to further improve the system's collaborative control capabilities in complex and dynamic scenarios. Additionally, reinforcement learning algorithms suitable for multi-agent settings with

variable input scales will be explored to enhance the system's adaptability to changing target quantities and overall robustness.

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

Funding: This research is supported by the National Natural Science Foundation of China (Grant No. 62271166 and 62401177).

Data Availability Statement: The raw data supporting the conclusions of this article will be made available by the authors on request. This study did not generate or use any human participant data. The images shown in Figure 15 were collected solely for hardware demonstration purposes, and all identifiable facial features were obscured to ensure anonymity. No raw human data was generated, collected, or analyzed in this work.

Conflicts of Interest: Author Yutong Jiang was employed by the company China North Vehicle Research Institute. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

References

1. Bhupathi, T.; Chittala, A.; Mani, V. A video surveillance based security model for military bases. In Proceedings of the 2021 International Conference on Recent Trends on Electronics, Information, Communication & Technology (RTEICT), Bangalore, India, 27–28 August 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 517–522.
2. Kim, D.; Kim, K.; Park, S. Automatic PTZ camera control based on deep-Q network in video surveillance system. In Proceedings of the 2019 International Conference on Electronics, Information, and Communication (ICEIC), Auckland, New Zealand, 22–25 January 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 1–3.
3. Wang, S.; Tian, Y.; Xu, Y. Automatic control of PTZ camera based on object detection and scene partition. In Proceedings of the 2015 IEEE International Conference on Signal Processing, Communications and Computing (ICSPCC), Ningbo, China, 19–22 September 2015; IEEE: Piscataway, NJ, USA, 2015; pp. 1–6.
4. Sharma, A.; Subasharan, V.; Gulati, M.; Wanniarachchi, D.; Misra, A. CollabCam: Collaborative Inference and Mixed-Resolution Imaging for Energy-Efficient Pervasive Vision. *ACM Trans. Internet Things* **2025**, *6*, 1–35. [CrossRef]
5. Zhou, X.; Li, X.; Zhu, Y.; Ma, C. Towards building digital twin: A computer vision enabled approach jointly using multi-camera and building information model. *Energy Build.* **2025**, *335*, 115523. [CrossRef]
6. Hu, K.; Li, M.; Song, Z.; Xu, K.; Xia, Q.; Sun, N.; Zhou, P.; Xia, M. A review of research on reinforcement learning algorithms for multi-agents. *Neurocomputing* **2024**, *599*, 128068. [CrossRef]
7. Liang, J.; Miao, H.; Li, K.; Tan, J.; Wang, X.; Luo, R.; Jiang, Y. A Review of Multi-Agent Reinforcement Learning Algorithms. *Electronics* **2025**, *14*, 820. [CrossRef]
8. Zhuang, H.; Lei, C.; Chen, Y.; Tan, X. Cooperative decision-making for mixed traffic at an unsignalized intersection based on multi-agent reinforcement learning. *Appl. Sci.* **2023**, *13*, 5018. [CrossRef]
9. Chen, Y.; Tu, Z.; Zhang, S.; Zhou, J.; Yang, C. A synchronous multi-agent reinforcement learning framework for UVMS grasping. *Ocean. Eng.* **2024**, *307*, 118155. [CrossRef]
10. Lee, K.M.; Ganapathi Subramanian, S.; Crowley, M. Investigation of independent reinforcement learning algorithms in multi-agent environments. *Front. Artif. Intell.* **2022**, *5*, 805823. [CrossRef]
11. Matignon, L.; Laurent, G.J.; Le Fort-Piat, N. Independent reinforcement learners in cooperative markov games: A survey regarding coordination problems. *Knowl. Eng. Rev.* **2012**, *27*, 1–31. [CrossRef]
12. Sunehag, P.; Lever, G.; Gruslys, A.; Czarnecki, W.M.; Zambaldi, V.; Jaderberg, M.; Lanctot, M.; Sonnerat, N.; Leibo, J.Z.; Tuyls, K.; et al. Value-decomposition networks for cooperative multi-agent learning. *arXiv* **2017**, arXiv:1706.05296.
13. Rashid, T.; Samvelyan, M.; De Witt, C.S.; Farquhar, G.; Foerster, J.; Whiteson, S. Monotonic value function factorisation for deep multi-agent reinforcement learning. *J. Mach. Learn. Res.* **2020**, *21*, 1–51.
14. Jiang, Z.; Chen, Y.; Wang, K.; Yang, B.; Song, G. A Graph-Based PPO Approach in Multi-UAV Navigation for Communication Coverage. *Int. J. Comput. Commun. Control* **2023**, *18*. [CrossRef]
15. Lowe, R.; Wu, Y.I.; Tamar, A.; Harb, J.; Pieter Abbeel, O.; Mordatch, I. Multi-agent actor-critic for mixed cooperative-competitive environments. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 6382–6393.

16. Yu, C.; Velu, A.; Vinitsky, E.; Gao, J.; Wang, Y.; Bayen, A.; Wu, Y. The surprising effectiveness of ppo in cooperative multi-agent games. *Adv. Neural Inf. Process. Syst.* **2022**, *35*, 24611–24624.
17. Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; Klimov, O. Proximal policy optimization algorithms. *arXiv* **2017**, arXiv:1707.06347. [CrossRef]
18. Liu, L.; Ustun, V.; Kumar, R. Leveraging Organizational Hierarchy to Simplify Reward Design in Cooperative Multi-agent Reinforcement Learning. In Proceedings of the The International FLAIRS Conference Proceedings, FL, USA, 19–21 May 2024; Volume 37.
19. Xu, J.; Zhong, F.; Wang, Y. Learning multi-agent coordination for enhancing target coverage in directional sensor networks. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 10053–10064.
20. Chen, Q.; Heydari, B. Adaptive Network Intervention for Complex Systems: A Hierarchical Graph Reinforcement Learning Approach. *J. Comput. Inf. Sci. Eng.* **2025**, *25*, 061006. [CrossRef]
21. Ci, H.; Liu, M.; Pan, X.; Zhong, F.; Wang, Y. Proactive multi-camera collaboration for 3d human pose estimation. *arXiv* **2023**, arXiv:2303.03767. [CrossRef]
22. Masihullah, S.; Kandaswamy, S. A Decentralized Collaborative Strategy for PTZ Camera Network Tracking System using Graph Learning: Assessing strategies for information sharing in a PTZ camera network for improving vehicle tracking, via agent-based simulations. In Proceedings of the 2022 5th International Conference on Mathematics and Statistics, Paris, France, 17–19 June 2022; pp. 59–65.
23. Darázs, B.; Bukovinszki, M.; Kósa, B.; Remeli, V.; Tihanyi, V. Comparison of Barrier Surveillance Algorithms for Directional Sensors and UAVs. *Sensors* **2024**, *24*, 4490. [CrossRef]
24. Hou, Y.; Leng, X.; Gedeon, T.; Zheng, L. Optimizing camera configurations for multi-view pedestrian detection. *arXiv* **2023**, arXiv:2312.02144. [CrossRef]
25. Kim, D.; Yang, C.M. Reinforcement learning-based multiple camera collaboration control scheme. In Proceedings of the 2022 Thirteenth International Conference on Ubiquitous and Future Networks (ICUFN), Barcelona, Spain, 5–8 July 2022; IEEE: Piscataway, NJ, USA; pp. 414–416.
26. Yin, M.; Sun, Z.; Guo, B.; Yu, Z. Effi-MAOT: A Communication-Efficient Multi-Camera Active Object Tracking. In Proceedings of the 2023 19th International Conference on Mobility, Sensing and Networking (MSN), Nanjing, China, 14–16 December 2023; IEEE: Piscataway, NJ, USA, 2023; pp. 9–16.
27. Li, J.; Xu, J.; Zhong, F.; Kong, X.; Qiao, Y.; Wang, Y. Pose-assisted multi-camera collaboration for active object tracking. *Proc. AAAI Conf. Artif. Intell.* **2020**, *34*, 759–766. [CrossRef]
28. Wang, P.; Ma, R.; Yang, Z.; Hao, Q. Robotic Camera Array Motion Planning for Multiple Human Face Tracking Based on Reinforcement Learning. *IEEE Sens. J.* **2024**, *24*, 24649–24658. [CrossRef]
29. Veesam, S.B.; Rao, B.T.; Begum, Z.; Patibandla, R.L.; Dcosta, A.A.; Bansal, S.; Prakash, K.; Faruque, M.R.I.; Al-Mugren, K. Multi-camera spatiotemporal deep learning framework for real-time abnormal behavior detection in dense urban environments. *Sci. Rep.* **2025**, *15*, 26813. [CrossRef] [PubMed]
30. Bustamante, A.L.; Molina, J.M.; Patricio, M.A. Distributed active-camera control architecture based on multi-agent systems. In Proceedings of the Highlights on Practical Applications of Agents and Multi-Agent Systems: 10th International Conference on Practical Applications of Agents and Multi-Agent Systems, Salamanca, Spain, 28–30 March 2012; Springer: Berlin, Germany, 2012; pp. 103–112.
31. Fang, H.; Liu, H.; Wen, J.; Yang, Z.; Li, J.; Han, Q. Automatic visual enhancement of PTZ camera based on reinforcement learning. *Neurocomputing* **2025**, *626*, 129531. [CrossRef]
32. Mnih, V.; Kavukcuoglu, K.; Silver, D.; Graves, A.; Antonoglou, I.; Wierstra, D.; Riedmiller, M. Playing atari with deep reinforcement learning. *arXiv* **2013**, arXiv:1312.5602. [CrossRef]
33. Wang, Y.; Sun, Y.; Liu, Z.; Sarma, S.E.; Bronstein, M.M.; Solomon, J.M. Dynamic graph cnn for learning on point clouds. *ACM Trans. Graph. (tog)* **2019**, *38*, 1–12. [CrossRef]
34. Kipf, T.N.; Welling, M. Semi-supervised classification with graph convolutional networks. *arXiv* **2016**, arXiv:1609.02907.
35. Ge, Z.; Liu, S.; Wang, F.; Li, Z.; Sun, J. Yolox: Exceeding yolo series in 2021. *arXiv* **2021**, arXiv:2107.08430. [CrossRef]
36. Chen, K.; Wang, J.; Pang, J.; Cao, Y.; Xiong, Y.; Li, X.; Sun, S.; Feng, W.; Liu, Z.; Xu, J.; et al. MMDetection: Open MMLab Detection Toolbox and Benchmark. *arXiv* **2019**, arXiv:1906.07155. [CrossRef]