drones

MDPI

Article

Enhancing Unmanned Aerial Vehicle Path Planning in Multi-Agent Reinforcement Learning through Adaptive Dimensionality Reduction

Haotian Shi [1,†], Zilin Zhao [2,†], Jiale Chen [3,†], Mengjie Zhou [4] and Yang Liu [1,*]

[1] College of Instrumentation and Electrical Engineering, Jilin University, Jilin 130061, China
[2] School of Software, Jilin University, Jilin 130015, China; zhaozl9921@mails.jlu.edu.cn
[3] School of Communication Engineering, Jilin University, Jilin 130012, China
[4] Department of Computer Science, University of Bristol, Bristol BS8 1QU, UK
* Correspondence: liu_yang@jlu.edu.cn
† These authors contributed equally to this work.

Abstract: Unmanned Aerial Vehicles (UAVs) have become increasingly important in various applications, including environmental monitoring, disaster response, and surveillance, due to their flexibility, efficiency, and ability to access hard-to-reach areas. Effective path planning for multiple UAVs exploring a target area is crucial for maximizing coverage and operational efficiency. This study presents a novel approach to optimizing collaborative navigation for UAVs using multi-agent reinforcement learning (MARL). To enhance the efficiency of this process, we introduce the Adaptive Dimensionality Reduction (ADR) framework, which includes Autoencoders (AEs) and Principal Component Analysis (PCA) for dimensionality reduction and feature extraction. The ADR framework significantly reduces computational complexity by simplifying high-dimensional state spaces while preserving crucial information. Additionally, we incorporate communication modules to facilitate inter-UAV coordination, further improving path planning efficiency. Our experimental results demonstrate that the proposed approach significantly enhances exploration performance and reduces computational complexity, showcasing the potential of combining MARL with ADR techniques for advanced UAV navigation in complex environments.

Keywords: multi-agent reinforcement learning (MARL); dimensionality reduction; unmanned aerial vehicles (UAVs)

check for updates

Citation: Shi, H.; Zhao, Z.; Chen, J.; Zhou, M.; Liu, Y. Enhancing Unmanned Aerial Vehicle Path Planning in Multi-Agent Reinforcement Learning through Adaptive Dimensionality Reduction. *Drones* **2024**, *8*, 521. https://doi.org/10.3390/drones8100521

Academic Editor: Oleg Yakimenko

Received: 15 August 2024
Revised: 18 September 2024
Accepted: 21 September 2024
Published: 25 September 2024

Copyright: © 2024 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) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Unmanned Aerial Vehicle (UAV) technology has made tremendous progress in recent years, with an increasingly wide range of applications, from military reconnaissance [1] to logistics [2] and distribution, and from agriculture and plant protection [3] to disaster relief [4]. The development of UAV technology has provided efficient and economical solutions [5] for these industries. UAVs are popular for monitoring complex terrain due to their high flexibility, mobility, and low deployment costs [6]. Simultaneously, utilizing multiple UAVs to explore target areas and establish a communication system for information sharing can significantly enhance exploration efficiency, reduce overall costs, and minimize the impact of individual UAV failures on terrain exploration. However, limited resources make informative path planning (IPP) a critical issue.

IPP methods include grid-based search, potential field methods, and heuristic algorithms like A* [7]. Grid-based search ensures thorough coverage by dividing the area into a grid and systematically exploring each cell, but it is computationally expensive and inefficient for large areas. Potential field methods create a virtual field where UAVs are attracted to the target and repelled by obstacles, resulting in smooth paths; however, they can become trapped in local minima. Heuristic algorithms such as A* provide optimal

paths by evaluating multiple potential routes and selecting the best one based on cost. Still, they struggle with dynamic and complex scenarios where environmental conditions constantly change.

Reinforcement learning (RL) [8] has emerged as a promising solution to address these limitations. RL enables UAVs to learn and adapt to dynamic environments by interacting with the environment and receiving feedback on their actions. This learning process allows for the development of optimal policies that improve over time, providing robust solutions to complex scenarios. However, traditional RL methods are primarily designed for single-agent systems and face significant challenges when extended to multi-agent scenarios, such as scalability and coordination among agents.

To overcome these challenges, multi-agent reinforcement learning (MARL) was developed [9,10], which extends RL to multiple interacting agents. MARL enables UAVs to handle dynamic and uncertain environments collaboratively by learning and adapting through interaction with the environment and each other. MARL allows for the development of optimal policies that improve over time and provide robust solutions to complex scenarios. This makes it particularly powerful in situations where traditional methods fall short. Additionally, MARL facilitates coordination among multiple UAVs, enhancing their collective performance. However, despite these advantages, MARL approaches can be computationally intensive and may suffer from efficiency and scalability issues when applied to large-scale multi-UAV systems.

In our method, images captured by the UAVs are first processed through a combination of Autoencoders (AEs) and Principal Component Analysis (PCA) as part of the Adaptive Dimensionality Reduction (ADR) framework. This preprocessing step significantly reduces the computational load by simplifying high-dimensional input data while preserving essential features. AEs are primarily used for feature extraction and handling complex data, while PCA provides an efficient means of further reducing the dimensionality, particularly in less complex scenarios. The output from the ADR framework is then fed into the actor network of the MARL framework, which is responsible for decision-making and path planning. Furthermore, we deploy communication modules that allow UAVs to share information and coordinate their actions effectively. This integrated approach not only enhances the overall efficiency of the system but also improves the performance and robustness of UAV navigation in complex and dynamic environments.

This paper focuses on active data collection using a team of UAVs for terrain monitoring scenarios. The objective is to create a map of an initially unknown, non-homogeneous binary target variable on a 2D terrain, such as identifying crop infestations in an agricultural setting or locating victims in a disaster scenario, utilizing image measurements captured by the UAVs. Figure 1 can more intuitively show the working scene of the drone. We address the challenge of multi-agent IPP, where we design information-rich paths for the UAVs to cooperatively gather sensor data while adhering to energy, time, or distance constraints. The goal is to enable the UAVs to dynamically monitor the terrain, concentrating on areas of interest with high information value.

The main contribution of this paper is the introduction of a novel multi-agent deep reinforcement learning-based IPP approach for adaptive terrain monitoring using UAV teams. Our approach supports decentralized on-board decision-making and achieves cooperative 3D path planning with variable team sizes. Our main contributions include the following:

1. Markov Modeling and Environment Modeling: We develop a comprehensive Markov model to represent the states of UAVS and the environment dynamics, enabling more accurate predictions and decision-making in complex terrains.

2. ADR framework: We introduce the ADR framework, which integrates AEs and PCA to preprocess image data captured by the UAVs. This dual-method approach allows for efficient dimensionality reduction and feature extraction, enhancing computational efficiency and improving the quality of the input for the reinforcement learning framework.

3. Implementation of Communication Modules: We establish robust communication protocols that allow UAVs to share information and coordinate their actions effectively. This enhances overall system performance, ensuring better coverage and data collection.

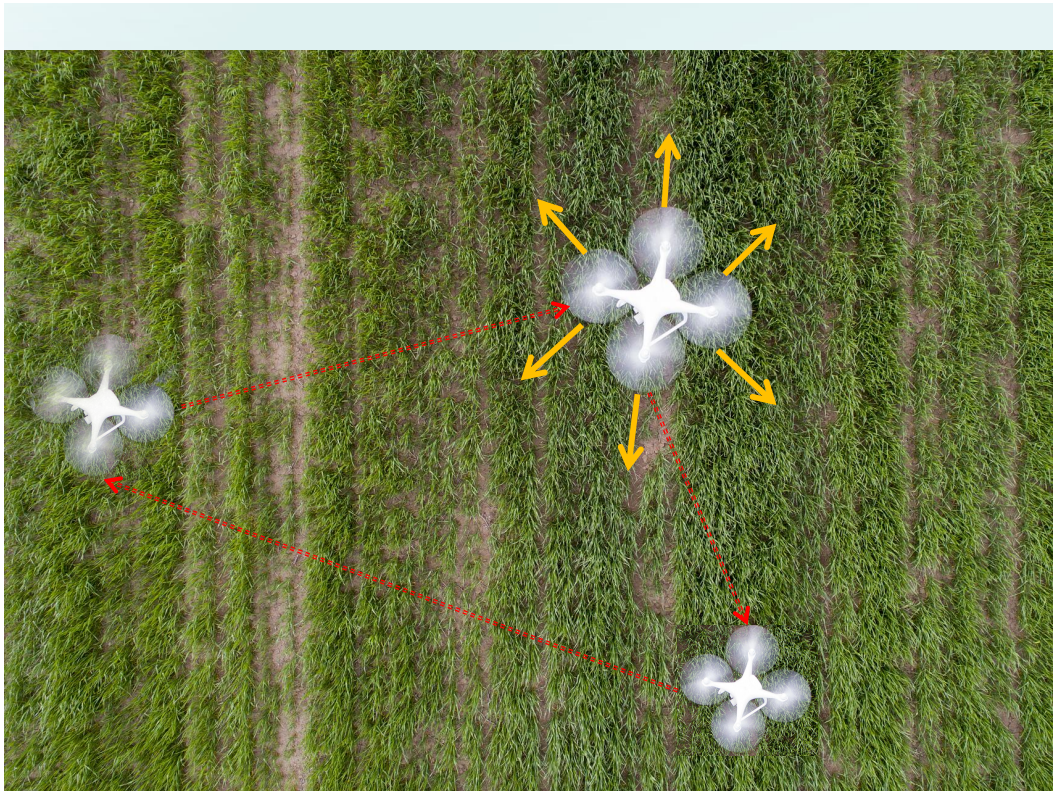


Figure 1. This figure illustrates a multi-UAV (Unmanned Aerial Vehicle) system operating cooperatively above a crop field. Three UAVs are shown flying over the crops. The red dashed arrows indicate the communication paths between the UAVs. Each UAV determines its next measurement location based on locally available information, as indicated by the yellow arrows showing possible action directions. The red arrows shows the communication channels among the UAVs which means they are cooperation instead of working isolatedly. This cooperative path planning enables the UAVs to effectively cover and monitor temperature variations or other parameters of interest in the field. In the figure, the size of the drones represents the different altitudes at which they fly.

Additionally, we address the credit assignment problem in cooperative IPP using counterfactual multi-agent policy gradients (COMA). Our approach significantly improves planning performance and computational efficiency, demonstrating its potential for UAV navigation in complex and dynamic environments.

The whole article includes 6 sections. Section 2 reviews the relevant literature, covering UAV IPP methods, RL, MARL, AE, and PCA. Section 3 details our proposed method, introducing the Markov model, our ADR framework, policy learning model, and inter-agent communication model. Mathematical formulations and figures are used to define our methodology. Section 4 outlines our experimental setup, including parameter configurations and justifications. We present heat maps of three environments, illustrating the significance of different regions. Section 5 analyzes results from various scenarios and examines factors contributing to the outcomes. Section 6 concludes by summarizing the ADR framework's contributions to MARL in UAV path planning and suggests future research directions.

2. Related Work

Traditional UAV IPP methods include the grid-based search proposed by Yamauchi [11], the potential field methods introduced by Khatib [12], and heuristic algorithms like A* de-

veloped by Hart, Nilsson, and Raphael [13]. Grid-based search ensures thorough coverage but is computationally expensive and inefficient for large areas. Potential field methods create smooth paths using virtual forces but often get trapped in local minima. The A* algorithm selects optimal paths by evaluating multiple potential routes but struggles in dynamic and complex scenarios. To address these limitations, LaValle [14] proposed the Rapidly-exploring Random Trees (RRT) algorithm, which efficiently explores large spaces but often produces non-smooth paths. The Particle Swarm Optimization (PSO) method, introduced by Kennedy and Eberhart [15], optimizes flight paths through iterative improvements but requires careful parameter tuning and is computationally intensive. The limitations of these traditional methods highlight the need for more advanced solutions, such as RL, to achieve more effective UAV path planning in dynamic and complex environments.

RL has also been widely applied to UAVs. Wei et al. [16] designed a constrained exploration and exploitation strategy using Q-networks, but this method is limited to single-agent intelligence. Vashisth et al. [17] proposed a dynamically constructed graph restricting agents to local planning actions, allowing for better path exploration. However, it cannot transfer policies to real robots without localization and perception uncertainties. Rueckin et al. [10] combined tree search with an offline-trained neural network, significantly improving information perception capabilities with small data sets. Pirinen et al. [7] introduced a strategy for using UAVs to find unknown target regions based on limited visual cues. Chen et al. [18] developed a graph-based deep RL method for exploration, selecting map frontiers that reduce map uncertainty and travel time. However, their approach is limited to 2D workspaces, whereas we consider 3D planning. Jonas Westheimer et al. [19] introduce new network feature representations to effectively learn path planning in a 3D workspace. However, their research faces significant efficiency issues as the amount of data increases.

MARL is a more promising direction for IPP as it is closer to real-world scenarios and introduces more complex environments and higher requirements. Foerster et al. [20] introduced counterfactual COMA to address the credit assignment problem. Lowe et al. [21] proposed the Multi-Agent Deep Deterministic Policy Gradient (MADDPG), which allows for centralized training and decentralized execution, improving performance in complex environments. Yousef et al. [22] proposed a novel mean-field flight resource allocation optimization method to minimize the Age of Information (AoI) of perceived data. This method successfully optimized the UAV flight trajectories and the data collection scheduling of ground sensors, showing significant improvements compared to DQN. Iqbal and Sha [23] proposed Independent Q-Learning (IQL), which simplifies the training process of multiple agents while maintaining high performance. However, the computational complexity and scalability issues of MARL in large-scale systems still require further research and solutions. To address this, recent studies such as Zhang et al. [24] have proposed a hierarchical MARL method to reduce computational burden and improve scalability through hierarchical decision-making. Additionally, Peng et al. [25] introduced a distributed MARL framework that reduces communication overhead and enhances system resilience, demonstrating potential applications in large-scale multi-agent systems.

In order to improve the training effect of the network, David et al. [26] proposed the autoencoder (AE) based on the back propagation of neuron-like unit networks. Hinton et al. [27] first proposed the use of gradient descent to fine-tune the weights of AE networks, reducing the data dimension to improve the training effect. Although the AE can effectively solve the problem of insufficient feature extraction and overfitting, there are also some problems, such as long training time and insufficient accuracy. In order to solve these problems, scholars have conducted research and made some improvements to the AE.

For image processing, the working principle of the AE requires the data to be transformed into a one-dimensional vector for post-processing, which leads to the loss of the two-dimensional structure information of the image. Therefore, Ranjan et al. [28] introduced convolutional neural networks to preserve two-dimensional spatial information by replacing the fully connected layers with convolution and pooling operations in them.

Zhao et al. [29] proposed image classification based on DSAE dimensionality reduction, but it causes the problem of gradient disappearance as the number of convolutional neural network (CNN) layers increases. To this end, they added a residual network module to the 3D CNN and proposed 3DDRN [30]. For samples of various sizes, DSAE can effectively extract low-dimensional features from the original image and perform well in classification. Guo et al. [31] performed feature extraction by adding a CNN to a stacked autoencoder (SAE). This method can achieve high accuracy through simple unsupervised pre-training and single supervised training, simplifying the complex calculation process. For the noise in the image, Revathi et al. [32] proposed an AE method based on DCNN to deal with the noise in the image, which effectively improved the accuracy.

To further exploit the potential of the AE, Cheng et al. [33] added a regularization term to the hidden neurons of a discriminative stacked autoencoder (DSAE) to bring similar samples closer together in the mapping space. At the same time, this approach can use the regularization optimization objective function to fine-tune the DSAE. Inspired by them, Wang et al. [34,35] conducted a study on the relationship between the number of neurons in the hidden layer and classification accuracy and proposed that for simple images, the gain is no longer obvious after the number of neurons is close to the input data dimension. However, for complex image processing, the classification progress will improve as the number of neurons increases.

PCA is another linear feature projection method used to reduce the dimensionality of data. It simplifies the classifier design and reduces the computational burden of pattern recognition technology by reducing the dimension and keeping most of the relevant features in the distant data. Qifa et al. [36] pointed out that the traditional singular value decomposition (SVD) is not well applicable when there are outliers and missing data in the measurement. They leveraged the sensitivity of PCA to outliers and applied iterated reweighted least squares (IRLS) to decompose each element, leading to the development of their L1-PCA approach. Chris et al. [37] used PCA to minimize the sum of squared errors and proposed rotation invariant L1-norm PCA (R1-PCA). This method guarantees that there is a unique global solution and the solution is rotation invariant. After comparison, R1-PCA can deal with outliers more effectively, and when extended to K-means clustering, L1-norm K-means performs worse, while the R1 K-means method is significantly better. However, PCA can only describe a good coordinate system for all feature distributions and does not consider class separation. Therefore, in order to improve the accuracy of recognition, it is necessary to provide high-level separable features. Matsumura et al. [38] used PCA technology to analyze EMG data, which effectively improved the recognition accuracy and recognition speed.

To further enhance the efficiency of MARL in UAV path planning, we propose an innovative approach incorporating ADR. By preprocessing the image data captured by the UAVs, the ADR performs dimensionality reduction and feature extraction, significantly reducing the computational burden of high-dimensional input data. The output from the autoencoders is then fed into the execution network of the MARL framework, thereby improving the overall system efficiency and performance and enhancing the robustness of UAV navigation in complex and dynamic environments.

3. Method

This section presents our proposed method. It covers these key components: the Markov Decision Process modeling, the ADR framework, the policy learning module, the Actor–Critic Network, and the Communication module.

3.1. Markov Decision Process Modeling

To address the problem of optimizing UAV collaborative navigation in multi-agent reinforcement learning, we model it as a Markov Decision Process (MDP). The MDP, as illustrated in Figure 2, serves as a mathematical framework to model decision-making in scenarios where outcomes are influenced by both random factors and the decisions made

by an agent. It has been demonstrated that the use of converting such problems to MDP is optimal [39].

State space. This encompasses all possible states where an agent can exist. We take a picture of the measurement range through the camera of the UAV and obtain the likelihood estimate z_i^t of UAV i projected onto the flat terrain at time t. Each UAV i stores a local posterior map confidence $\mathcal{M}^{t+1}$. According to the measurement value of UAV i and the altitude value, its position p_i^t can be obtained. We define $S_t = \{\mathcal{M}^t, \boldsymbol{p}_{1:N}^t, b\}$ as the state of the global environment at time t, where $p_{1:N}^t$ represents the current location, and b represents the remaining mission budget.

Action space. This comprises all possible actions an agent can take to influence the environment. Each agent takes an action of a fixed step size in a 3D discrete environment, which includes east, south, west, north, up, and down. They can only move within a prescribed discrete location grid P and cannot move outside the environment.

Reward function. Map state-action pairs to real numbers and provide feedback to the agent by assigning a numerical reward for each action taken in a given state.

$$R^t\left(s^t, u^t, s^{t+1}\right) = \alpha \frac{H\left(\mathcal{M}^{t+1}\right) - H\left(\mathcal{M}^t\right)}{H\left(\mathcal{M}^t\right)} + \beta, \tag{1}$$

where u^t represents the joint action of all UAVs, and $\mathcal{M}^t$ is the map state at time t. The reduction in entropy from the current map state to the next, $\mathcal{M}^{t+1}$, is rewarded, as it reflects a decrease in environmental uncertainty, indicating that the UAVs have gathered more valuable information and improved task completion. In the reward function, the parameter α controls the weight of the entropy reduction, influencing the focus on reducing uncertainty. The parameter β, as a bias term, adjusts the baseline reward, encouraging behaviors such as exploring new areas or maintaining specific flight paths.

γ is the discount factor, a value between 0 and 1, that determines the importance of future rewards, balancing short-term and long-term gains.

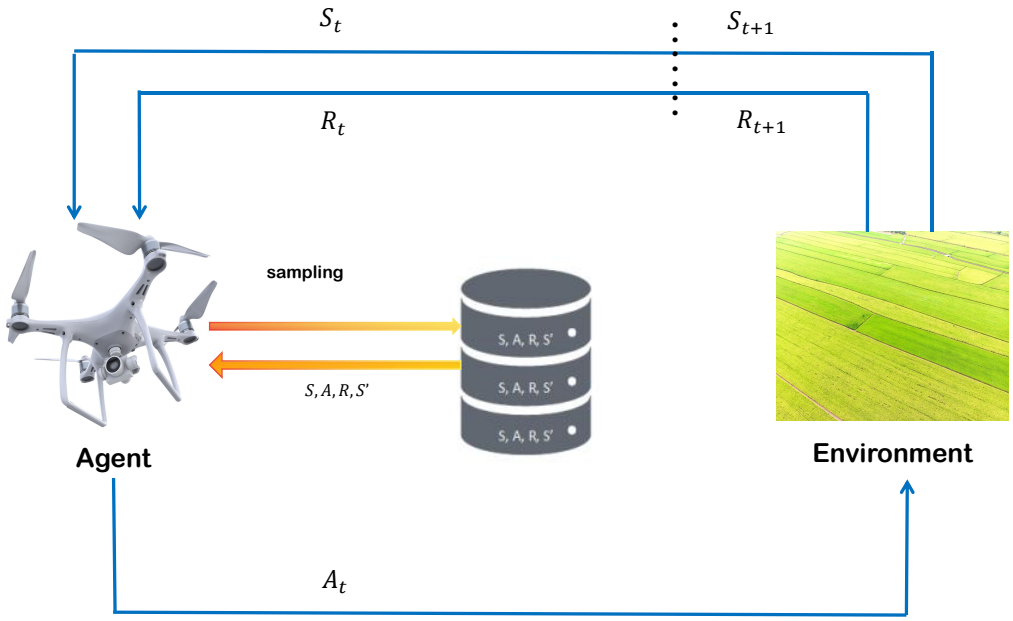


Figure 2. This diagram illustrates the workflow of an MDP where a drone (Agent) interacts with the environment. At time t, the drone is in state S_t and takes action A_t. The environment then responds with a new state S_{t+1} and a reward R_t. These data (S, A, R, S') are stored and sampled to help improve the decision-making strategy of the drone.

3.2. The Proposed Framework

In this section, we present the proposed framework designed for optimizing multi-UAV collaboration in MARL tasks. The framework consists of three key modules: the ADR module, the policy learning module, and the communication module. First, the ADR module performs dimensionality reduction on high-dimensional image data, which is then used in the policy learning module to guide UAV decision-making through an actor–critic architecture. The communication module enables efficient data exchange and coordination between UAVs, allowing for real-time collaboration.

3.2.1. The ADR Framework

The ADR framework, which includes AE and PCA, is specifically designed to optimize data processing efficiency and accuracy in MARL. In our work, the ADR framework is applied to a multi-UAV collaborative task where drones equipped with RGB cameras scan specific terrains, and the framework manages the data flow from the cameras to the Action Network.

The AE achieves data encoding and decoding through two main components:

- *Encoder:* $z = f(x)$, where x is the input data.
- *Decoder:* $\hat{x} = g(z)$, where z is the encoded latent representation.

In processing image data, AE uses this structure to extract key features, aiming to improve model robustness to environmental changes. Standard references on autoencoders, such as the work by [34], discuss in detail the effects of autoencoders in dimension reduction and feature learning.

In our experiments, the AE was deployed as shown in Figure 3:

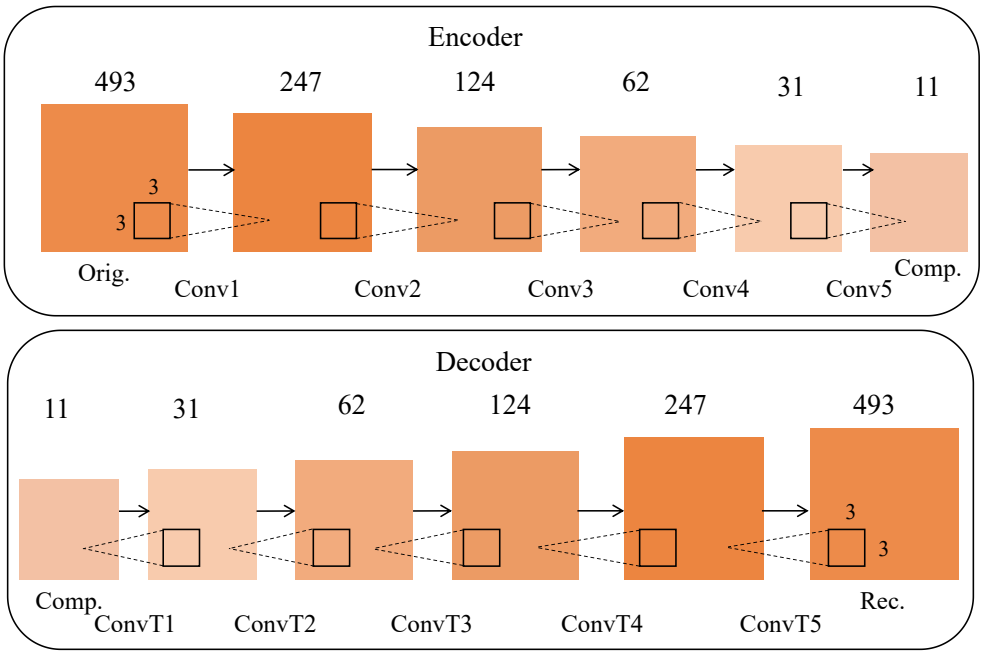


Figure 3. This is the structure of the AE, where all convolutional kernels use a size of 3×3.

Specifically, we tested three data processing methods: none, AE, and PCA, in that order. During the 'none' method, 6000 493×493 images were randomly sampled using UAV cameras to train the AE and PCA.

For the AE, it is divided into an encoder and a decoder. The 493×493 images are compressed into an 11×11 matrix through multiple convolutions by the encoder and then magnified back to the original image size by the decoder in a symmetric inverse process, known as reconstruction. By optimizing the mean squared error (MSE) between

the reconstructed and original images, the AE network parameters are adjusted until the reconstruction image closely matches the original image.

At this point, we extract the encoder part of the network, directly use it to generalize the processing of camera data in actual MARL tasks, and compress the data before submitting it to the actor network.

PCA reduces dimensions by linearly transforming the original data matrix X into $Y = XP$, where P is the matrix of principal components extracted from the covariance matrix of the data.

Ref. [37] provides a thorough overview of PCA techniques, particularly emphasizing their application in multivariate data analysis. The integration of these theories and methods, particularly when combined with AE technology within our ADR framework, is intended to improve terrain scanning and feature extraction, potentially offering a viable strategy for managing complex data in practical scenarios.

In our experiments, we organized the 6000 493 $\times$ 493 images sampled in the 'none' method into a large data matrix X, where each row represents a flattened, camera-captured original image. We first applied mean centering on X to ensure the mean of each feature is zero. Next, we calculated the covariance matrix C and performed eigen decomposition to extract the principal components. Finally, based on the input shape required by the actor network, we selected the 121 principal components with the highest eigenvalues to form matrix P. In subsequent MARL tasks, the camera data X is multiplied by P to compress the dimensions to 11 $\times$ 11 before being submitted to the actor network.

3.2.2. Policy Learning Module

In this section, we introduce the policy learning module, which plays a critical role in enabling cooperative behavior among UAVs for effective path planning and decision-making. The objective of this module is to guide the UAVs in dynamically adapting to various environmental conditions while ensuring optimal coordination. By leveraging reinforcement learning techniques, specifically the actor–critic framework, the policy learning module allows UAVs to learn and execute robust strategies in real time.

As illustrated in Figure 4, the overall architecture is designed to facilitate cooperative behavior among UAVs by combining dimensionality reduction and reinforcement learning. Each UAV collects sensor data from its environment, generating a sensor map which is then processed into a local map. To ensure computational efficiency, the ADR framework is applied to these local maps, performing dimensionality reduction and extracting relevant features from the high-dimensional image data. After this step, the reduced information is passed into the actor network, where a policy is generated to guide each UAV's actions. These actions are performed during the deployment phase, allowing the UAVs to make decisions independently based on their local observations.

In the training phase, the critic network evaluates the actions taken by the UAVs using a global map that provides an overall view of the environment. The critic network computes the Q-values for the actions and feeds this information back to the actor network to optimize the policy. Through this combination of dimensionality reduction, policy learning, and centralized evaluation, the architecture enables efficient, cooperative UAV operations across different environments.

The structure of the actor network is shown in Figure 5. The actor network utilizes a deep convolutional neural network (CNN) architecture to process multiple input data sources. The network consists of three convolutional layers (Conv1, Conv2, Conv3), followed by fully connected layers (FC layers). Each convolutional layer is followed by a ReLU activation function, and the number of filters increases with each layer, allowing the network to capture complex spatial relationships in the input data. After the final convolutional layer, the output is flattened and passed through the fully connected layers (FC1, FC2, FC3), gradually reducing the feature dimensions before generating the UAV's possible actions. The final output layer contains 6 nodes, representing the specific actions

the UAV can take based on the processed information, guiding its behavior during the deployment phase.

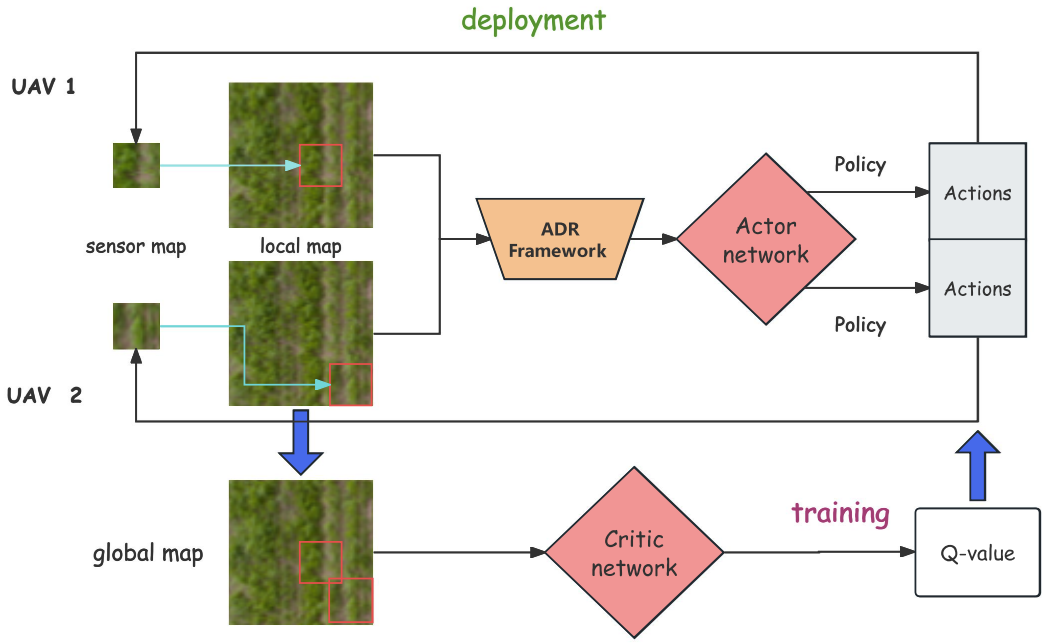


Figure 4. The architecture includes auto dimensionality reduction (ADR), policy generation (actor network), and evaluation (critic network) for UAV coordination during deployment and training phases.

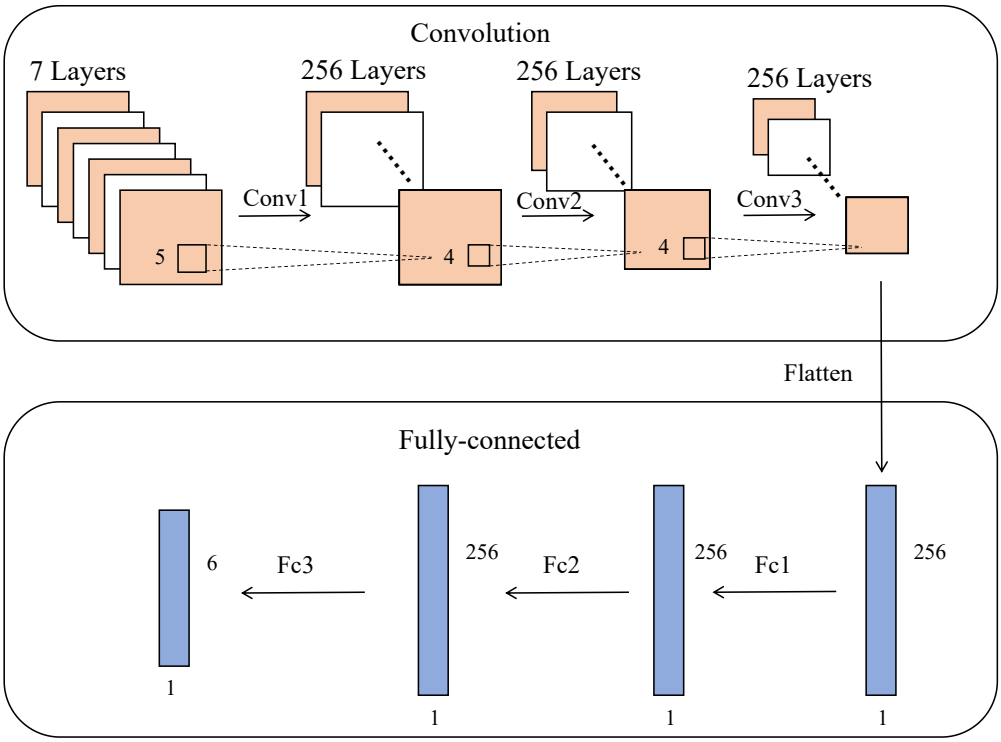


Figure 5. This is the structure of the actor network, which consists of two main parts: one is a convolutional network and the other is a fully connected network. The structure of the critic network is similar to that of the actor network.

The actor network processes various pieces of input information, including the UAV's identifier and remaining mission budget, a local map centered on the UAV's position, the weighted entropy of the local map to provide environmental uncertainty information, the

weighted entropy of the UAV's measurements, and a footprint map indicating the areas observed by all UAVs within communication range. These inputs are processed by the actor network to generate effective strategies, enabling the UAVs to collaborate and achieve the mission objectives. The image data among these inputs are processed by the ADR module to ensure that key information is retained while reducing dimensionality.

The structure of the critic network is similar to that of the actor network, with both networks sharing the same core architecture. However, compared to the actor network, the critic network processes additional input information. In addition to the inputs received by the actor network, the critic network further incorporates global environment information. These additional inputs include a global position map encoding the positions of all agents, the global map state, the weighted entropy of the global map, the map cells currently spanned by all agents' fields of view, and the actions of other agents.

Next, SCOMAP (State-Compressed Multi-Agent Policy Gradients) serves as the core algorithm to optimize the actions of each agent through the actor and critic networks. SCOMAP builds on the COMA framework but integrates state compression to enhance computational efficiency and scalability. The centralized critic network evaluates the joint state–action values of all agents by utilizing compressed state representations, which are output from the ADR module.

The critic network receives additional global environment information, including the global map state, the positions of all agents, and their actions, as mentioned earlier. This allows the critic network to compute the advantage function using counterfactual baselines, which measure the impact of each agent's action by comparing it to the baseline actions of other agents. This method ensures that each agent is credited for its individual contribution to the team's performance while accounting for the actions of others.

$$A^i(s^t, u^t) = Q_\pi(s^t, u^t) - \sum_{u_i'^t \in U} \pi(u_i'^t | \omega_i^t) Q_\pi(s^t, (u_i'^t, u_{-i}^t)). \tag{2}$$

This function compares the joint state–action value to the baseline, which is computed by marginalizing over all possible actions of the agent while keeping the actions of the other agents constant. By using state compression, SCOMAP achieves more efficient policy updates and optimized cooperation among UAVs. More detailed equations and derivations can be found in Ref. [20].

3.2.3. Communication Module

To improve the efficiency of multi-agent reinforcement learning (MARL) in UAV systems, we developed a communication module that operates within a limited range. This module enables UAVs to exchange field-of-view information with nearby agents, thereby enhancing collaboration and task execution. The communication module allows each UAV to share critical environmental and status data within a specified communication range, facilitating real-time decision-making. As shown in Figure 6, each UAV is equipped with a communication system and an RGB camera sensor to capture image data reflecting its current state. The communication module consists of key components, the interaction diagram between the modules is shown in Figure 7, including the data transmission and data processing units, which enable efficient data exchange and fusion.

The data transmission unit ensures low-latency and high-bandwidth data transmission within the communication range. It allows UAVs to share their current state information, such as position and speed, and environmental data when they are within communication range. This unit also handles potential random communication failures, ensuring that critical data are transmitted efficiently. On the other hand, the data processing unit receives field-of-view data from other UAVs and processes them, extracting meaningful information to update each UAV's global situational awareness, which is crucial for optimizing path planning and decision-making.

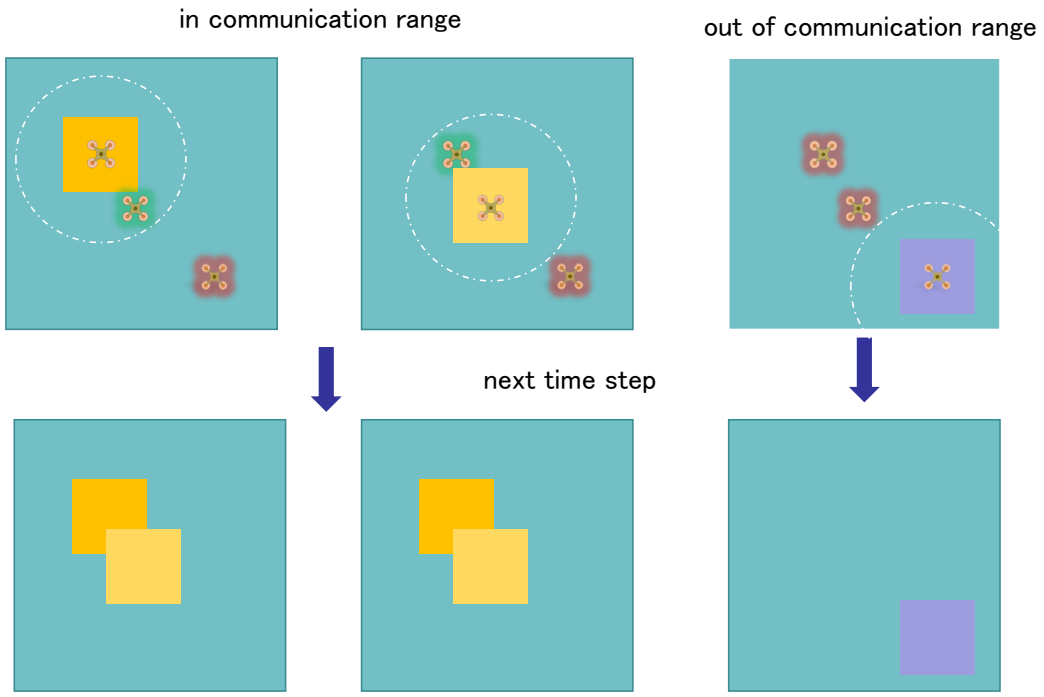


Figure 6. Illustration of the internal communication of the UAV. The UAVs interact with each other about their current measurements (shown here as orange, yellow, and purple rectangles). They must be within the communication range to interact, and the range is drawn in white circles in the figure. The green background represents in range, and the red represents out of range.

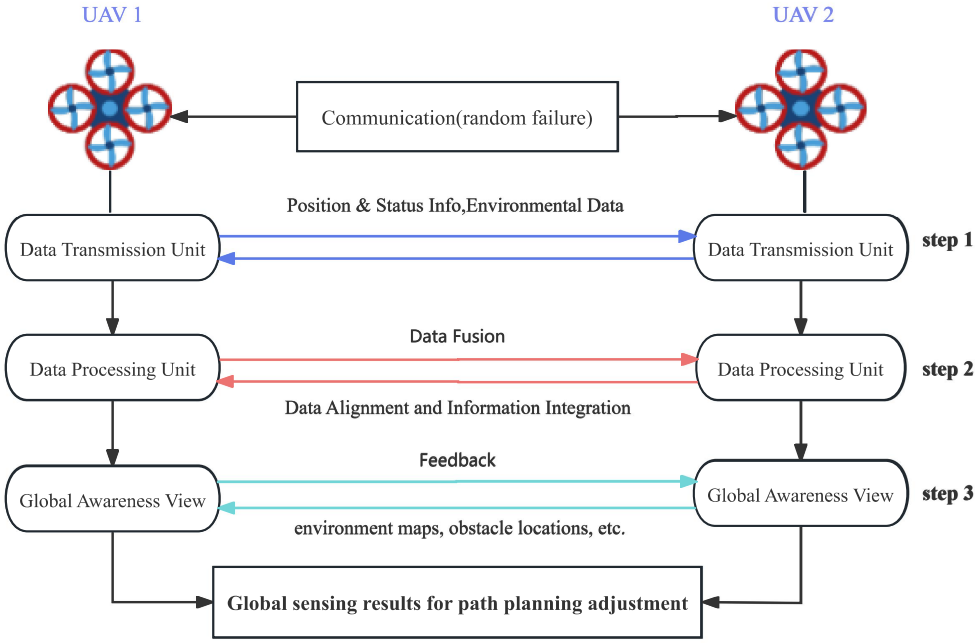


Figure 7. The communication module enables UAVs to exchange position, status, and environmental data for collaboration. The data transmission unit (step 1) sends this information to nearby UAVs, which is then processed in the data processing unit (step 2). The data are fused and aligned, providing a global awareness view (step 3), which includes environment maps and obstacle locations. The integrated global sensing results are used to adjust each UAV's path planning.

In the simulation environment, each UAV periodically broadcasts its map data and key observations to other UAVs within its communication range. These shared data include image data captured by the camera, which are used to extract the UAV's current state, such as altitude, speed, and direction, as well as environmental information. Specifically, the information shared between UAVs can be categorized into two types: position and status information, which includes the current position, flight speed, direction, and altitude of each UAV, and environmental observation information, which includes detailed map data collected via virtual sensors. The latter type of information includes terrain details, obstacles, and other significant environmental features, allowing each UAV to sense its operational environment and adjust its flight trajectory to optimize task efficiency and ensure safety.

Upon receiving the field-of-view data from other UAVs within the communication range, the data processing unit in the communications module performs data fusion. This process involves aligning data from different UAVs to ensure temporal and spatial consistency. Given that UAVs might be in different locations and data transmission may experience delays, the data are aligned based on timestamps and position information to mitigate errors caused by time lag. This alignment ensures that all the UAV information can be combined on a unified temporal and spatial framework. After alignment, the field-of-view data from multiple UAVs are integrated using a weighted averaging method, which assigns weights based on communication quality, field coverage scope, and data accuracy. This integration produces a comprehensive global environmental view, combining multiple observations for a more accurate and complete understanding of the environment.

The integrated data provide each UAV with a global view of the environment, significantly enhancing situational awareness and enabling real-time decision-making. This global view includes a detailed map of the terrain, the locations of obstacles, and the positions and statuses of other UAVs. Based on this global information, each UAV can dynamically adjust its flight path to avoid potential risks and optimize task execution. The integrated global awareness data are then fed back into the UAV's path planning system to improve mission performance, allowing UAVs to collaborate more effectively in a multi-agent environment and address complex flight tasks and environmental challenges.

4. Experiment

In this study, we used Python 3.8 to develop the experimental code, with key libraries including pygame for graphical interface support, flake8 for code style checks, and pytest for unit testing.

To verify that drones can efficiently collaborate in different scenarios, we conducted experiments in three distinct environments. These scenarios are as follows: Scenario 1—a strip area occupying a certain percentage of the space, Scenario 2—a rectangular area located in the corner of the scene, and Scenario 3—a strip area in the middle. The images of Scenarios 1, 2, and 3 are presented in Figures 8–10, respectively. We identified real terrains matching these shapes and created heatmaps based on the areas of interest. In these heatmaps, the dark (especially red) parts are recognized as areas of high interest. During the mission, drones need to collect as much information as possible from these high-interest areas.

In the experiments, we focused on the separate training processes of AE and PCA, using large-scale image data with a resolution of 493×493 pixels captured by UAVs for dimensionality reduction to improve computational efficiency and reduce data complexity. The AE was trained on 6000 images captured by the UAVs, with a batch size of 128, using the Adam optimizer with an initial learning rate set to 0.001, for a total of 100 epochs. On the other hand, PCA used Incremental Principal Component Analysis (IPCA) to process the same data, reducing them to 121 principal components. Given the large data volume, IPCA employed a batch size of 4096 and utilized incremental fitting to manage the computational demands of large-scale data processing.

For each experiment, we executed 50 terrain monitoring missions in a 50 m by 50 m area with a map resolution of 10 cm. The planning resolution was set to 10 m, and the flight altitude was limited to between 20 m and 30 m. We used a camera with a 60-degree field of view, ensuring adjacent measurements do not overlap when taken from the lowest altitude. Considering increased sensor noise at higher altitudes, we simulated noise probabilities of 0.98, 0.75, and 0.625 for altitudes of 5, 10, and 15 m, respectively. The UAV team consisted of 4 agents with a communication radius of 25 m.

The experimental environment included a seed value of 3 to ensure varied start positions for each UAV. The sensor was an RGB camera with a resolution of 10 cm (57 pixels in both x and y directions). The simulation used a random field with a cluster radius of 5 m. The mapping prior was set to 0.5. The experiment included constraints such as a spacing of 5 m, a minimum altitude of 5 m, a maximum altitude of 15 m, and a budget of 14 actions. UAVs have a maximum speed of 5 m per second, a maximum acceleration of 2 m per second squared, and a sampling time of 2 s. Missions were conducted using the COMA method in training mode, with 1500 episodes and a patience of 100. The remaining relevant hyperparameters are presented in Table 1.

In all experiments, we primarily measured the performance of the algorithms using rewards, supplemented by Q-values and entropy to further evaluate the experimental outcomes. The reward value directly reflects the effectiveness of a strategy in completing tasks within a specific environment and serves as the core metric for evaluating the quality of the strategy. It clearly indicates whether the strategy meets the expected goals, which is particularly crucial in complex scenarios. The Q-value assesses the long-term value of the strategy, offering insights into future decisions; a high Q-value suggests that the strategy is effective not only in the current step but also in multiple future steps, which is essential for the long-term optimization of path planning. Entropy evaluates the stability and diversity of the strategy; low entropy indicates that the strategy is becoming stable, while high entropy may suggest that the strategy is still exploring different decision paths. By considering the reward, Q-value, and entropy together, a more comprehensive understanding of the overall performance of the algorithm can be achieved, ensuring more stable and efficient path planning in practical applications. Our experimental results were averaged over ten trials, and the plotted curves reflect these averages, with the shaded areas representing the variance across the experiments.

Table 1. Parameter configuration table.

Parameter Name	Value	Meaning
γ	0.99	Discount factor for future rewards
λ	0.8	Weighting factor for future rewards
α	0.2	Parameter in the reward function
β	0.5	Parameter in the reward function
τ	0.01	Factor for soft update of target network parameters
learning_rate	0.00001	Step size for gradient descent
momentum	0.9	Factor to accelerate gradient descent in the relevant direction
gradient_norm	10	Maximum norm for gradients
data_passes	5	Number of times the dataset is passed through
batch_size	600	Number of samples processed before the model is updated
eps_max	0.5	Maximum value for ϵ in ϵ-greedy strategy, controlling the exploration rate
eps_min	0.02	Minimum value for ϵ in ϵ-greedy strategy, controlling the exploration rate
sampling_time	2	Sampling time in seconds; determines how often the data are collected

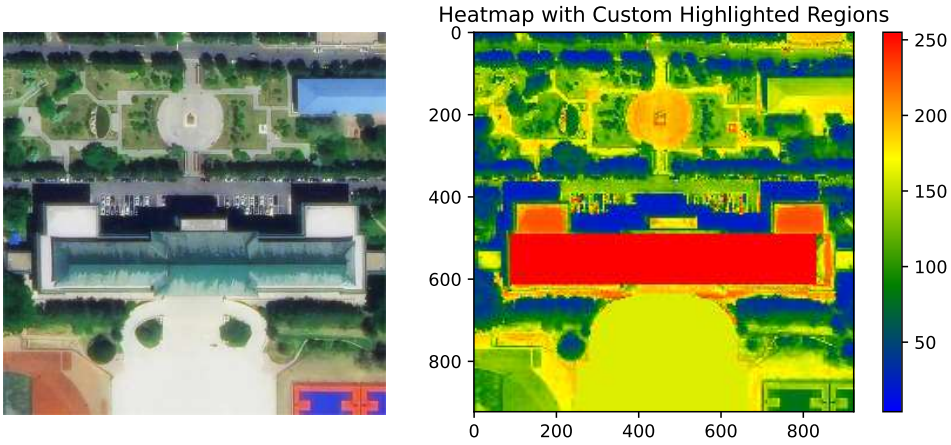


Figure 8. This is an aerial view of the Chaoyang Campus of Jilin University. The green rectangular building in the center, called the "Geology Palace", is our area of interest, matching the description of our Scenario 0. Besides the central rectangular building, the two smaller buildings next to it (shown in orange in the image) are also considered secondary areas of interest.

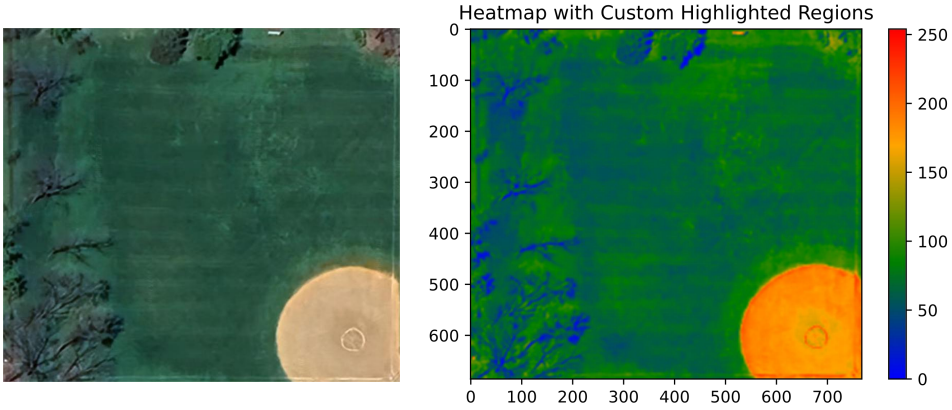


Figure 9. This is an aerial view of a softball field. The quarter-circle area in the corner is our area of interest, matching the description of our Scenario 1. In this image, we do not designate the highest-interest area (red); only secondary areas of interest are marked. This is because, in real-life scenarios, it is not always possible for every scene to have a highest-interest area.

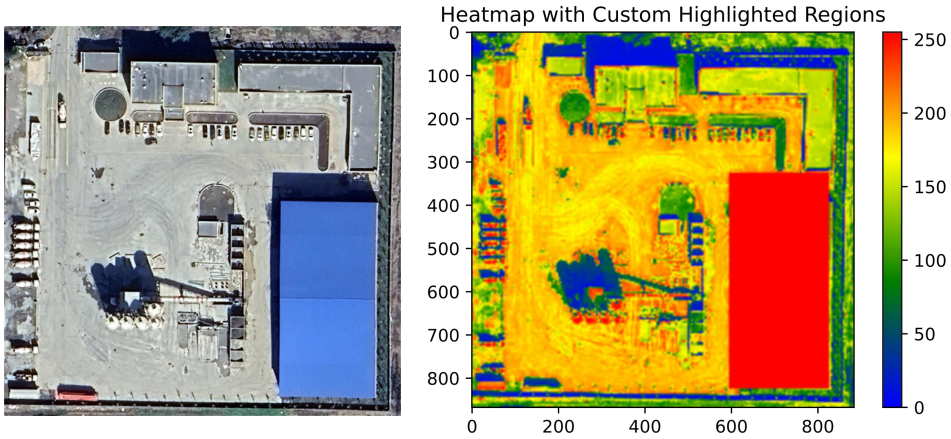


Figure 10. This is an aerial view of a construction site. The blue rectangular building in the corner is our area of interest, matching the description of Scenario 2. In this image, besides the blue rectangular building being the highest-interest area, we designate many secondary areas of interest to simulate the real-life scenario where a scene contains numerous areas of interest.

5. Result

This section presents a comprehensive analysis of experimental results across three distinct scenarios. We evaluate different methods using multiple metrics including reward, entropy, and Q-value. Additionally, we provide an examination of the factors contributing to each set of outcomes, offering insights into the effectiveness of our proposed method in diverse situations.

5.1. Result in Scenario 1

In the multi-agent reinforcement learning task of Scenario 1, different dimensionality reduction strategies have a significant impact on model performance, and the reward trend is presented in Figure 11. Overall, the AE strategy performed the best throughout the training process, ultimately achieving the highest reward value of approximately 0.45, demonstrating strong stability and learning effectiveness. At the beginning of training, the PCA (red line) method slightly outperforms both AE (green line) and the method without dimensionality reduction (blue line), suggesting that PCA has certain advantages in the initial exploration of strategies. However, the slightly fluctuating performance of AE may be attributed to the fact that the dimensionality reduction process introduces additional complexity, requiring more time to adapt to the environment.

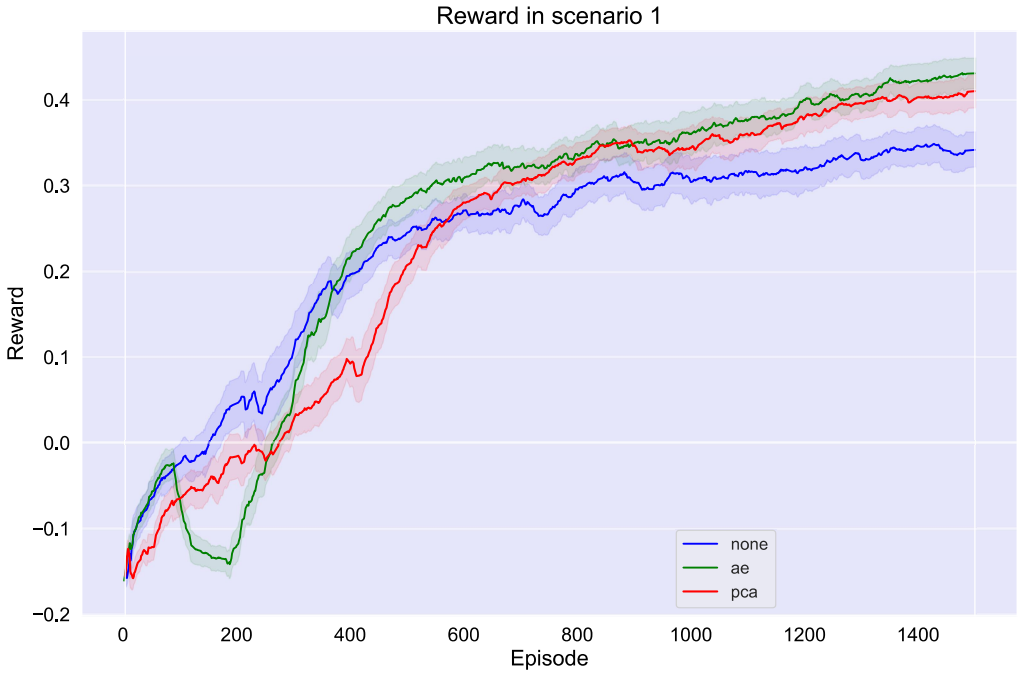


Figure 11. Reward trend: training performance across different dimensionality reduction strategies in Scenario 1.

While PCA performs better than AE and the non-reduced strategy during the first 200 training rounds, its performance becomes more erratic as training progresses, failing to maintain a consistent advantage. Consequently, the final reward value for PCA is slightly lower than that of AE, around 0.4. The strategy without dimensionality reduction performs the worst in the initial stages and exhibits slow growth, with a final reward value of only about 0.35. This outcome suggests that in complex scenarios, appropriate dimensionality reduction is crucial for enhancing the effectiveness of multi-agent reinforcement learning. Specifically, the AE method is more effective in capturing key features within the scene, thereby improving overall strategy effectiveness.

In the Q-value performance of Scenario 1, as shown in Figure 12, the PCA method (red line) performs the best in the later stages of training, with a final Q-value close to 2.0, showing its advantage in long-term strategy optimization. The AE (green line) has the next best performance, with a Q-value stabilized at around 1.8. The dimensionalized method (blue line), on the other hand, although performing moderately well in the early stages, has a lower final Q value that stays at around 1.5, slightly inferior to the other two dimensionalized strategies. We can see that in the later stages of training, the AE (green line) approach ends up with the lowest entropy value of the three, converging to 0.2. This suggests that the strategies with dimensionality reduction using AE exhibit the highest level of determinism, i.e., a more focused and stable choice of strategies. In contrast, the final entropy values of the PCA (red line) and no dimensionality reduction (blue line) methods are slightly higher, approaching 0.25, implying that their strategies are slightly less deterministic and still retain a degree of exploratory or uncertainty. Therefore, AE not only exhibits a better reward value on the results of training but also possesses a higher strategy certainty, indicating that AE can capture and utilize the key features in the scene more efficiently, helping the intelligence to converge to the optimal strategy faster and more stably.

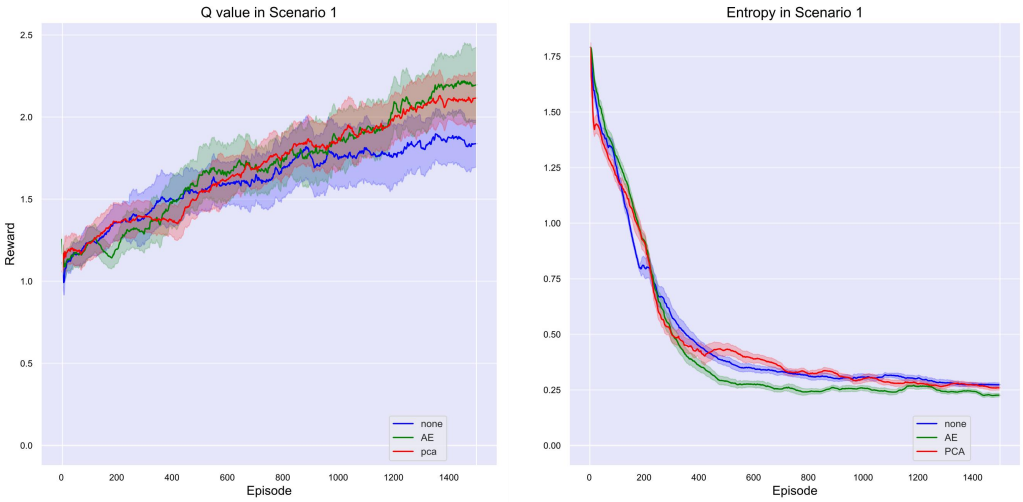


Figure 12. Q-value and entropy trends: comparative analysis of dimensionality reduction strategies in Scenario 1.

5.2. Result in Scenario 2

As shown in Figure 13, the results based on dimensionality reduction in the multi-agent reinforcement learning process indicate that both the AE and PCA slightly outperform the approach without dimensionality reduction (None), but the overall improvement is not significant, likely due to the lower complexity of the maps being processed. As observed in the heat map, performance in the region of sub-interest in the lower right corner is limited, likely due to the low significance of that region. In the reward results graph, while AE and PCA exhibit marginally better stability and higher reward values during the middle and later stages of training, the differences compared to the approach without dimensionality reduction are not substantial.

Overall, Figure 14 shows that while AE and PCA provide slight enhancements to algorithm performance, they do not fully leverage their potential benefits in this scenario, with the low map complexity likely being a key factor in this outcome. From the Q-value and entropy results graphs, it is evident that the model processed by AE performs relatively better in terms of Q-value, particularly in the middle and late stages of training, where the Q-value for AE is noticeably higher than that of the other two methods. This suggests that AE effectively enhances strategy quality during the process. Meanwhile, the model processed by PCA shows a quicker reduction in entropy, indicating that PCA facilitates

faster convergence to a more stable strategy. However, despite these slight advantages in different metrics, the Q-values and entropy of all three methods eventually converge to similar fixed values, suggesting that in this particular scenario, the overall performance of the methods becomes comparable, likely due to the low complexity of the scenario, which limits the potential advantages of dimensionality reduction methods.

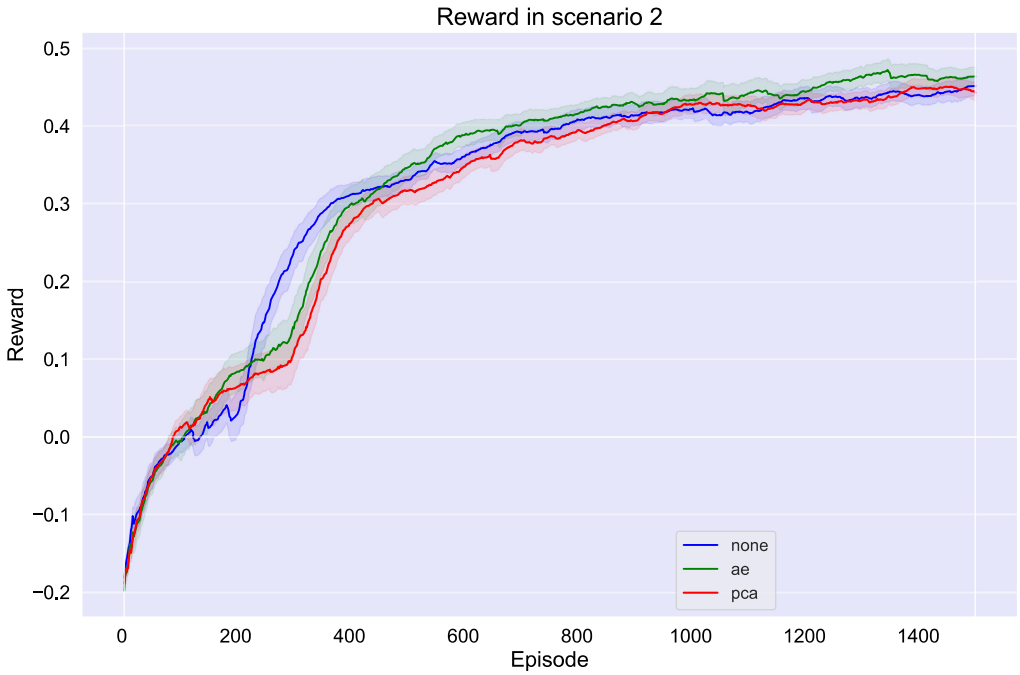


Figure 13. Reward trend: training performance across different dimensionality reduction strategies in Scenario 2.

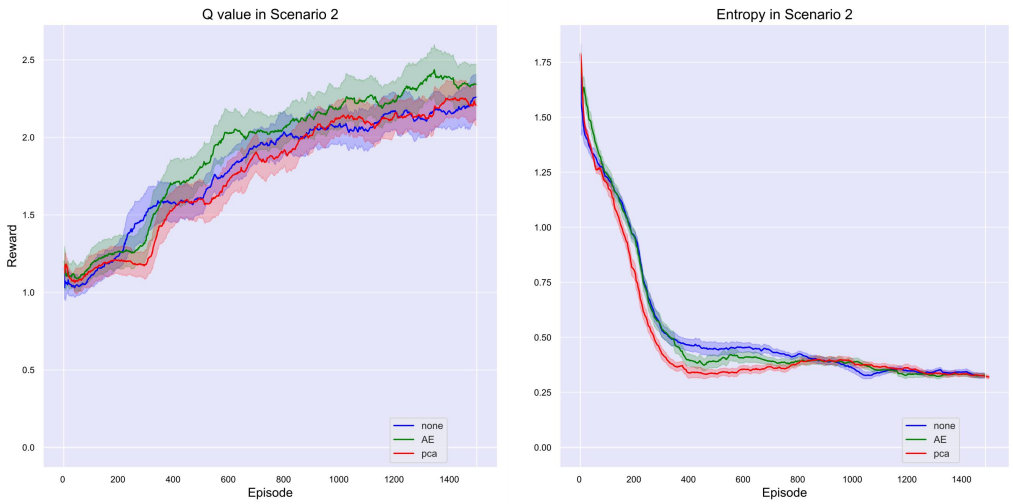


Figure 14. Q-value and entropy trends: comparative analysis of dimensionality reduction strategies in Scenario 2.

5.3. Result in Scenario 3

In the third scenario, as illustrated in Figure 15 regarding rewards, although PCA ultimately slightly surpassed the AE, it is important to note that the AE consistently led during the early stages of training. Specifically, the AE maintained higher reward values throughout the early and middle stages of training compared to PCA and the no-dimensionality-reduction method (None), indicating that the AE was able to extract effective features more quickly and enhance strategy quality at a faster rate. However, as

training progressed, PCA gradually caught up and slightly outperformed the AE in the later stages. This shift may be attributed to the ability of PCA to better manage various features as the complexity of the scenario increases, thereby improving overall model performance. Nonetheless, the AE exhibited strong performance throughout most of the training, demonstrating its advantage in handling complex data during the initial phases. Overall, while PCA held a slight edge in final performance, the AE performed exceptionally well across the entire training process, particularly in the early stages, with faster convergence and greater stability. This suggests that different dimensionality reduction methods may offer distinct advantages at various stages of training, and the final choice should take into account the specific application scenario and model requirements.

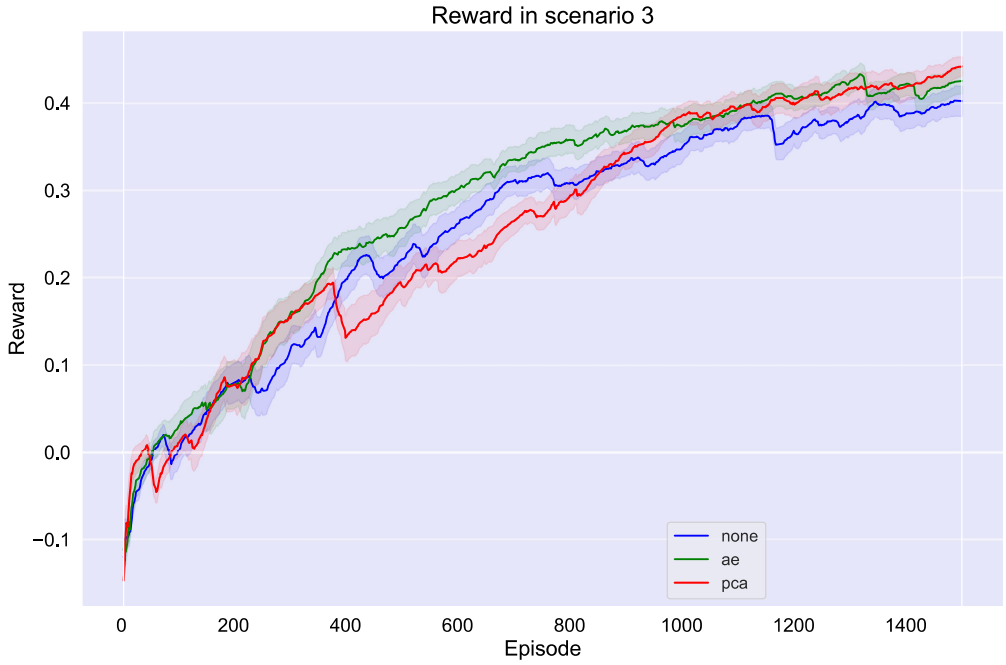


Figure 15. Reward trend: training performance across different dimensionality reduction strategies in Scenario 3.

From the Q-value graph in Figure 16, it is evident that the AE performed exceptionally well during the early stages of training, quickly raising the Q-values and demonstrating its ability to extract key features and optimize strategy quality in the initial phase. However, as training progressed, PCA gradually caught up with the AE and showed greater stability in the later stages, eventually slightly surpassing the AE. This further confirms that in complex scenarios, PCA is better able to extract diverse features and enhance strategy quality over longer training periods.

From the entropy graph in Figure 16, it can be seen that the model processed by PCA exhibited a faster decrease in entropy during training, indicating quicker convergence to a stable strategy. This aligns with earlier observations, suggesting that PCA can more effectively reduce uncertainty when handling complex data and stabilize the strategy in a shorter amount of time. Conversely, the entropy graph shows that the method without dimensionality reduction had a slower decrease in entropy, and the final converged entropy value was slightly higher than those of the AE and PCA. This indicates that without dimensionality reduction, the stability of the model strategy was somewhat lower.

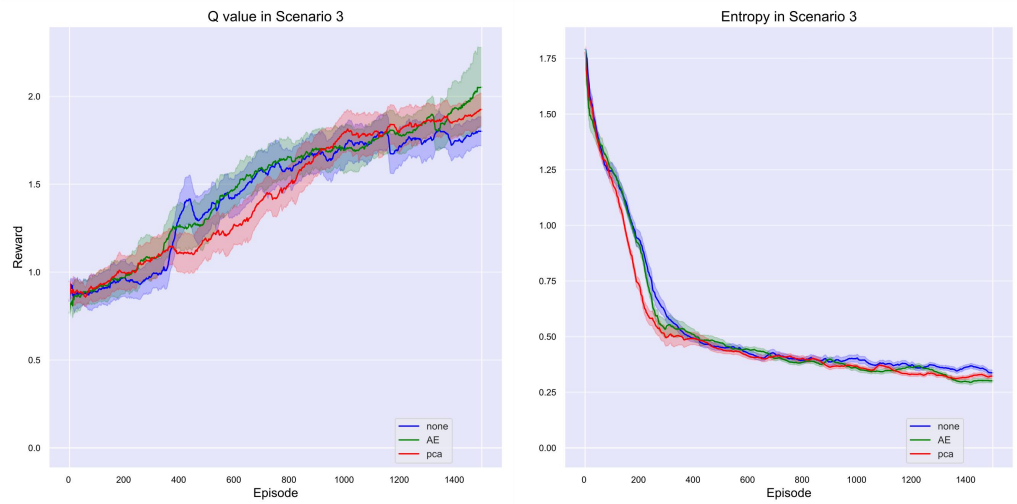


Figure 16. Q-value and entropy trends: comparative analysis of dimensionality reduction strategies in Scenario 3.

6. Conclusions

In this study, we proposed and explored the application of the ADR framework in multi-agent reinforcement learning, particularly focusing on its impact on UAV path planning tasks. The ADR framework optimizes system performance by flexibly selecting appropriate dimensionality reduction methods, such as the AE or PCA, to enhance computational efficiency and strategy stability.

This framework effectively handles complex, high-dimensional data while reducing data dimensionality in a way that preserves essential information, thereby lowering computational complexity and accelerating the training and exploration processes of the model. It provides a flexible and efficient dimensionality reduction solution for multi-agent systems, suitable for various task requirements and environmental complexities. By intelligently selecting the most suitable dimensionality reduction method, ADR significantly enhances the efficiency and performance of multi-agent reinforcement learning. Currently, the ADR framework primarily selects dimensionality reduction methods based on predefined scenario complexities. Future research could introduce more intelligent adaptive strategies, enabling the system to analyze environmental complexity and task requirements in real time and dynamically adjust the choice of dimensionality reduction methods, thereby further improving the adaptability and robustness of the system. Additionally, while this study focuses on UAV path planning, the ADR framework could extend to other fields, such as autonomous driving, intelligent surveillance, and robotic collaboration. Exploring the performance of ADR in these areas and its potential for broader applications will be an important direction for future research.

Author Contributions: Data curation, H.S. and J.C.; Formal analysis, H.S. and Z.Z.; Funding acquisition, Y.L.; Methodology, H.S., Z.Z. and M.Z.; Project administration, M.Z. and Y.L.; Resources, H.S.; Software, H.S., Z.Z. and J.C.; Supervision, M.Z. and Y.L.; Validation, H.S., Z.Z. and J.C. All authors have read and agreed to the published version of the manuscript.

Funding: This work is supported by the 2023 FAW Independent Innovation (Key CoreTechnology RD) Major Science and Technology Project, Project Number20230301002ZD, and the National College Student Innovation and Entrepreneurship Project of the College of Instrumentation and Electrical Engineering , Jilin University, Project Number 202410183295.

Data Availability Statement: Dataset available on request from the authors.

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

Abbreviations

The following abbreviations are used in this manuscript:

UAV	Unmanned Aerial Vehicle
IPP	informative path planning
RL	Reinforcement Learning
MARL	Multi-Agent Reinforcement Learning
AEs	Autoencoders
PCA	Principal Component Analysis
ADR	Adaptive Dimensionality Reduction
COMA	counterfactual multi-agent policy gradient
MDP	Markov Decision Process

References

1. Wang, Y.; Bai, P.; Liang, X.; Wang, W.; Zhang, J.; Fu, Q. Reconnaissance mission conducted by uav swarms based on distributed pso path planning algorithms. *IEEE Access* **2019**, *7*, 105086–105099. [CrossRef]
2. Estrada, M.A.R.; Ndoma, A. The uses of unmanned aerial vehicles–uav's-(or drones) in social logistic: Natural disasters response and humanitarian relief aid. *Procedia Comput. Sci.* **2019**, *149*, 375–383. [CrossRef]
3. Huang, Y.; Thomson, S.J.; Hoffmann, W.C.; Lan, Y.; Fritz, B.K. Development and prospect of unmanned aerial vehicle technologies for agricultural production management. *Int. J. Agric. Biol. Eng.* **2013**, *6*, 1–10.
4. Hildmann, H.; Kovacs, E. Using unmanned aerial vehicles (uavs) as mobile sensing platforms (msps) for disaster response, civil security and public safety. *Drones* **2019**, *3*, 59. [CrossRef]
5. Gupta, L.; Jain, R.; Vaszkun, G. Survey of important issues in uav communication networks. *IEEE Commun. Surv. Tutor.* **2015**, *18*, 1123–1152. [CrossRef]
6. Jawhar, I.; Mohamed, N.; Al-Jaroodi, J.; Agrawal, D.P.; Zhang, S. Communication and networking of uav-based systems: Classification and associated architectures. *J. Netw. Comput. Appl.* **2017**, *84*, 93–108. [CrossRef]
7. Pirinen, A.; Samuelsson, A.; Backsund, J.; Åström, K. Aerial view localization with reinforcement learning: Towards emulating search-and-rescue. *arXiv* **2022**, arXiv:2209.03694.
8. Nguyen, T.T.; Nguyen, N.D.; Nahavandi, S. Deep reinforcement learning for multiagent systems: A review of challenges, solutions, and applications. *IEEE Trans. Cybern.* **2020**, *50*, 3826–3839. [CrossRef]
9. Cui, J.; Liu, Y.; Nallanathan, A. Multi-agent reinforcement learning-based resource allocation for uav networks. *IEEE Trans. Wirel. Commun.* **2019**, *19*, 729–743. [CrossRef]
10. Rueckin, J.; Jin, L.; Popovic, M. Adaptive informative path planning using deep reinforcement learning for uav-based active sensing. In Proceedings of the 2022 IEEE International Conference on Robotics and Automation (ICRA 2022), IEEE International Conference on Robotics and Automation (ICRA), Philadelphia, PA, USA, 23–27 May 2022; pp. 4473–4479.
11. Yamauchi, B. A frontier-based approach for autonomous exploration. In Proceedings of the 1997 IEEE International Symposium on Computational Intelligence in Robotics and Automation CIRA'97, 'Towards New Computational Principles for Robotics and Automation', Monterey, CA, USA, 10–11 July 1997; pp. 146–151.
12. Khatib, O. Real-time obstacle avoidance for manipulators and mobile robots. *Int. J. Robot. Res.* **1986**, *5*, 90–98. [CrossRef]
13. Hart, P.E.; Nilsson, N.J.; Raphael, B. A formal basis for the heuristic determination of minimum cost paths. *IEEE Trans. Syst. Sci. Cybern.* **1968**, *4*, 100–107. [CrossRef]
14. LaValle, S.; Hutchinson, S. Optimal motion planning for multiple robots having independent goals. *IEEE Trans. Robot. Autom.* **1998**, *14*, 912–925. [CrossRef]
15. Shi, Y.; Eberhart, R. A modified particle swarm optimizer. In Proceedings of the 1998 IEEE International Conference on Evolutionary Computation, IEEE World Congress on Computational Intelligence (Cat. No.98TH8360), Anchorage, AK, USA, 4–9 May 1998; pp. 69–73.
16. Wei, Y.; Zheng, R. Informative path planning for mobile sensing with reinforcement learning. In Proceedings of the IEEE INFOCOM 2020—IEEE Conference on Computer Communications, Ser. IEEE INFOCOM, 39th IEEE International Conference on Computer Communications (IEEE INFOCOM), Electr Network, Toronto, ON, Canada, 6–9 July 2020; pp. 864–873.
17. Vashisth, A.; Ruckin, J.; Magistri, F.; Stachniss, C.; Popovic, M. Deep reinforcement learning with dynamic graphs for adaptive informative path planning. *IEEE Robot. Autom. Lett.* **2024**, *9*, 1–8. [CrossRef]
18. Chen, F.; Martin, J.D.; Huang, Y.; Wang, J.; Englot, B. Autonomous exploration under uncertainty via deep reinforcement learning on graphs. In Proceedings of the 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Las Vegas, NV, USA, 24 October 2020–24 January 2021; pp. 6140–6147.
19. Westheider, J.; Rückin, J.; Popović, M. Multi-uav adaptive path planning using deep reinforcement learning. In Proceedings of the 2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Detroit, MI, USA, 1–5 October 2023; pp. 649–656.

20. Foerster, J.; Farquhar, G.; Afouras, T.; Nardelli, N.; Whiteson, S. Counterfactual multi-agent policy gradients. In Proceedings of the AAAI Conference on Artificial Intelligence, New Orleans, LA, USA, 2–7 February 2018; Volume 32. Available online: https://ojs.aaai.org/index.php/AAAI/article/view/11794 (accessed on 13 August 2024).
21. Lowe, R.; Wu, Y.I.; Tamar, A.; Harb, J.; Abbeel, O.P.; Mordatch, I. Multi-agent actor-critic for mixed cooperative-competitive environments. *Adv. Neural Inf. Process. Syst.* **2017**, *30* .
22. Emami, Y.; Gao, H.; Li, K.; Almeida, L.; Tovar, E.; Han, Z. Age of information minimization using multi-agent uavs based on ai-enhanced mean field resource allocation. *IEEE Trans. Veh. Technol.* **2024**, *73*, 13368–13380. [CrossRef]
23. Iqbal, S.; Sha, F. Actor-attention-critic for multi-agent reinforcement learning. In Proceedings of the 36th International Conference on Machine Learning, Long Beach, CA, USA, 9–15 June 2019; Ser. Proceedings of Machine Learning Research; Chaudhuri, K., Salakhutdinov, R., Eds.; Volume 97, pp. 2961–2970. Available online: https://proceedings.mlr.press/v97/iqbal19a.html (accessed on 13 August 2024).
24. Zhang, K.; Yang, Z.; Başar, T. Multi-agent reinforcement learning: A selective overview of theories and algorithms. In *Handbook of Reinforcement Learning and Control*; Springer: Cham, Switzerland, 2021; pp. 321–384.
25. Guo, Z.; Chen, Z.; Liu, P.; Luo, J.; Yang, X.; Sun, X. Multi-agent reinforcement learning-based distributed channel access for next generation wireless networks. *IEEE J. Sel. Areas Commun.* **2022**, *40*, 1587–1599. [CrossRef]
26. Rumelhart, D.E.; Hinton, G.E.; Williams, R.J. Learning representations by back-propagating errors. *Nature* **1986**, *323*, 533–536. [CrossRef]
27. Hinton, G.E.; Salakhutdinov, R.R. Reducing the dimensionality of data with neural networks. *Science* **2006**, *313*, 504–507. [CrossRef]
28. Ranjan, R.; Patel, V.M.; Chellappa, R. Hyperface: A deep multi-task learning framework for face detection, landmark localization, pose estimation, and gender recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2019**, *41*, 121–135. [CrossRef]
29. Zhao, J.; Hu, L.; Dong, Y.; Huang, L.; Weng, S.; Zhang, D. A combination method of stacked autoencoder and 3d deep residual network for hyperspectral image classification. *Int. J. Appl. Earth Obs. Geoinf.* **2021**, *102*, 102459. [CrossRef]
30. Kong, F.; Zhao, S.; Li, Y.; Li, D.; Zhou, Y. A residual network framework based on weighted feature channels for multispectral image compression. *Ad Hoc Netw.* **2020**, *107*, 102272. [CrossRef]
31. Guo, J.; Li, Y.; Dong, S.; Zhang, W. Innovative method of crop classification for hyperspectral images combining stacked autoencoder and CNN. *Trans. Chin. Soc. Agric. Mach.* **2021**, *52*, 225–232.
32. Revathi, K.; Ananth, D.P.B.J. A novel video compression then restoration artifact reduction method based on optical flow consistency to improve the quality of the video. *Eur. J. Mol. Clin. Med.* **2020**, *7*, 5463–5474.
33. Cheng, G.; Zhou, P.; Han, J. Duplex metric learning for image set classification. *IEEE Trans. Image Process.* **2018**, *27*, 281–292. [CrossRef]
34. Wang, Y.; Yao, H.; Zhao, S. Auto-encoder based dimensionality reduction. *Neurocomputing* **2016**, *184*, 232–242. [CrossRef]
35. Deng, L. The mnist database of handwritten digit images for machine learning research [best of the web]. *IEEE Signal Process. Mag.* **2012**, *29*, 141–142. [CrossRef]
36. Ke, Q.; Kanade, T. Robust *L*1 norm factorization in the presence of outliers and missing data by alternative convex programming. In Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), San Diego, CA, USA, 20–25 June 2005; Volume 1, pp. 739–746.
37. Ding, C.; Zhou, D.; He, X.; Zha, H. R1-PCA: Rotational invariant l 1-norm principal component analysis for robust subspace factorization. In Proceedings of the 23rd International Conference on Machine Learning—ICML'06, Pittsburgh, PA, USA, 25–29 June 2006; ACM Press: New York, NY, USA, 2006.
38. Matsumura, Y.; Fukumi, M.; Mitsukura, Y. Hybrid EMG recognition system by MDA and PCA. In Proceedings of the 2006 IEEE International Joint Conference on Neural Network, Vancouver, BC, Canada, 16–21 July 2006; pp. 5294–5300.
39. Puterman, M.L. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*; John Wiley & Sons: Hoboken, NJ, USA, 2014.

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.