2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

Trajectory Planning for UAV-Based Smart Farming Using Imitation-Based Triple Deep Q-Learning

Wencan Mao
National Institute of Informatics
Tokyo, Japan
wencan_mao@nii.ac.jp

Quanxi Zhou
The University of Tokyo
Tokyo, Japan
usainzhou@g.ecc.u-tokyo.ac.jp

Tomás Couso Coddou
Pontificia Universidad Católica de Chile
National Center for Artificial Intelligence
Santiago, Chile
tcouso@uc.cl

Manabu Tsukada
The University of Tokyo
Tokyo, Japan
mtsukada@g.ecc.u-tokyo.ac.jp

Yunling Liu
China Agricultural University
Beijing, China
liuyunling@cau.edu.cn

Yusheng Ji
National Institute of Informatics
Tokyo, Japan
kei@nii.ac.jp

***Abstract*—Unmanned aerial vehicles (UAVs) have emerged as a promising auxiliary platform for smart agriculture, capable of simultaneously performing weed detection, recognition, and data collection from wireless sensors. However, trajectory planning for UAV-based smart agriculture is challenging due to the high uncertainty of the environment, partial observations, and limited battery capacity of UAVs. To address these issues, we formulate the trajectory planning problem as a Markov decision process (MDP) and leverage multi-agent reinforcement learning (MARL) to solve it. Furthermore, we propose a novel imitation-based triple deep Q-network (ITDQN) algorithm, which employs an elite imitation mechanism to reduce exploration costs and utilizes a mediator Q-network over a double deep Q-network (DDQN) to accelerate and stabilize training and improve performance. Experimental results in both simulated and real-world environments demonstrate the effectiveness of our solution. Moreover, our proposed ITDQN outperforms DDQN by 4.43% in weed recognition rate and 6.94% in data collection rate.**

***Index Terms*—Trajectory Planning, Unmanned Aerial Vehicle (UAV), Smart Farming, Multi-Agent Reinforcement Learning (MARL), Deep Q-Network (DQN).**

I. Introduction

Smart farming represents an innovative approach that integrates information and communication technology into the cyber-physical farm management cycle [1]. Cutting-edge technologies, such as autonomous systems, image processing, machine learning, big data, cloud/edge computing, and wireless sensor networks, have emerged to drive this advancement, paving the way for a better and healthier agricultural practice that features increased production quantity and quality, reduced costs and labor efforts, and lowered fuel, fertilizer, and pesticide utilization [2].

This work is supported by JST ASPIRE (JPMJAP2325), JSPS KAKENHI Grant No. JP24K02937, JST SPRING GX (JPMJSP2108), and National Center for Artificial Intelligence CENIA FB210017, Basal ANID.
The first two authors contributed equally to this research.
Corresponding author: Yunling Liu (liuyunling@cau.edu.cn).

While traditional farming relies hugely on human labor, autonomous systems will be able to control actuators effectively, improve the utility, control resource usage, and ensure products conform to market requirements [3]. In particular, unmanned aerial vehicles (UAVs) have been incorporated into smart farming to provide imagery analysis, agricultural surveillance, and in-depth situation awareness. Such UAVs are mounted with lightweight cameras, and their applications include insecticide and fertilizer prospecting and spraying, seed planting, weed recognition, fertility assessment, mapping, and crop forecasting [4].

Under these circumstances, trajectory planning for UAV-based smart farming becomes a critical and challenging issue, especially for large-scale farmlands, where multiple UAVs are supposed to collaborate. First of all, due to the uncertainty in environmental factors, the distribution and status of the monitoring or managing targets (e.g., seed, crop, weed, and pest) have high spatio-temporal diversity. Without knowledge of such conditions, the effectiveness and efficiency of the smart farming practice will be significantly degraded. Second, the UAV-mounted cameras have a limited field of view (FoV), namely the maximum area that can be captured by a camera. When the size of the farmland area increases, the UAVs will only have partial observations, meaning that they can only receive incomplete or uncertain information about the environment and the actions of other UAVs. This adds complexity to the decision-making of each individual UAV and their collaboration with each other. Besides, the UAVs could have concurrent tasks, such as weed detection, recognition, and data collection from wireless sensors. Such tasks need to be completed simultaneously with their respective requirements. To address the above challenges, we formulate the trajectory planning problem as a Markov decision process (MDP) and deploy multi-agent reinforcement learning (MARL) to plan the trajectories for UAV-based smart farming.

979-8-3315-8160-2/26/$31.00 ©2026 IEEE

MARL simulates the interaction among multiple agents and the environment, where each agent is a UAV and the environment refers to the farmland. Unlike traditional optimization methods (e.g., linear programming, A* algorithm, and hyper-heuristics), MARL enables time-efficient policy learning and adaptation to uncertainty in the environment, including the distribution of weeds and wireless sensors, as well as the starting points of the UAVs. Furthermore, traditional MARL methods, such as Deep Q-Network (DQN), suffer from low scalability and efficiency. When the size of the field, the amount of weeds, and the number of wireless sensors increase, suboptimal policies can lead to poor task performance or excessive battery consumption. To address this issue, we propose an *imitation-based triple deep Q-network (ITDQN)*, with a novel design of an *elite imitation mechanism* together with a *mediator Q-network (mid Q-network)* on top of a double deep Q-network (DDQN). The elite imitation mechanism enables the agents to learn from high-performing individuals, thereby reducing exploration costs. Meanwhile, a mediator Q-network is incorporated between the online and target Q-networks, enhancing training performance, efficiency, and stability.

The main contributions of this work are listed as follows:

- We target a UAV-based smart farming problem, where UAVs are mounted with lightweight cameras for weed detection and recognition, while collecting information from wireless sensors. We formulate the trajectory planning problem as an MDP to address the high uncertainty in the environment, partial observations of UAVs, and the limited battery of UAVs.
- We propose a novel MARL-based algorithm, referred to as ITDQN, for trajectory planning for UAV-based smart farming. Our proposed ITDQN leverages an elite imitation mechanism to lower exploration costs, while incorporating a mediator Q-Network over DDQN to enhance the performance, efficiency, and stability of training.
- Extensive evaluations in both simulated and real-world environments validate the effectiveness of our approach. Moreover, the proposed ITDQN consistently outperforms DQN, DDQN, and heuristic baselines.

II. Related Works

A. Trajectory Planning for UAV-Based Smart Farming

Recent advances in UAV path planning for precision agriculture span methods targeting efficiency, scalability, and adaptability. Machine learning has been used for energy-efficient flight planning to reduce power consumption while preserving coverage [5]. Coverage path planning remains widely studied but scales poorly to large fields where exhaustive scanning becomes impractical [6], [7]. To lower computational cost, evolutionary approaches such as grey wolf optimization offer efficient alternatives to traditional optimization [8]. Deep learning-based adaptive planning (e.g., YOLOv4 detection with Monte Carlo dropout) improves robustness to uncertainty, though deterministic policies may limit the usage for multi-UAV [9]. Reinforcement learning (RL) has also shown promise, including DQN with bidirectional LSTMs for pest control [10] and Q-learning methods prioritizing shortest-distance routes that outperform A* and Dijkstra in single-UAV settings [11]. Overall, these works demonstrate the value of learning and optimization for UAV monitoring, while highlighting the need for scalable multi-agent planning.

B. Q-Learning, DQN, DDQN, and Variants

Q-learning [12] is a famous model-free RL algorithm for UAV-based smart farming that learns the optimal action-value function via iterative Bellman updates. However, classical Q-learning is limited to small, discrete state-action spaces, since tabular representations become infeasible in high-dimensional settings [13]. To improve scalability, Mnih et al. [14] proposed DQN, which approximates the Q-function with deep neural networks and stabilizes training using experience replay and a target network. To mitigate DQN's overestimation bias, Van Hasselt et al. [15] introduced DDQN, which decouples action selection and evaluation by selecting actions with the online network while evaluating them with the target network, improving stability and performance.

Building upon DQN/DDQN, several extensions were proposed to improve efficiency, exploration, and generalization. Dueling DQN [16] decomposes the Q-function into a state-value component and an advantage component, enabling more robust value estimation in states where action choices have similar outcomes. Prioritized experience replay (PER) [17] replaces uniform sampling with a priority-based scheme, favoring transitions with larger temporal-difference (TD) errors and thus accelerating learning. Noisy DQN [18] introduces parameterized noise into network weights, replacing ϵ-greedy exploration with adaptive, learnable stochasticity. Distributional RL approaches such as C51 [19], QR-DQN [20], and IQN [21] shift from modeling expected returns to full return distributions, providing richer training signals and empirically stronger performance. Rainbow DQN [22] integrates key methods, including DDQN, dueling networks, PER, noisy nets, distributional RL, and n-step returns, into a unified architecture that achieves state-of-the-art results. Other extensions include ES-DQN [23], which improves the bias-variance trade-off of TD targets, and DRQN [24], which incorporates recurrent networks for partial observability.

As another extension of DDQN, we propose a novel ITDQN to improve efficiency, exploration, and stability in trajectory planning for UAV-based smart farming.

III. UAV-Based Smart Farming

A. Exemplary Scenario

Due to the side effects and environmental harm of herbicides, reduction of the amounts of herbicides used in farmland is a critical step towards sustainable agriculture [25]. In conventional weed control, herbicides are sprayed uniformly across the farmland, treating the soil, crops, and weeds in the same manner. However, this measure will lead to the over-provisioning of herbicides and neglect the difference in herbicide sensitivity for different types of weeds. By

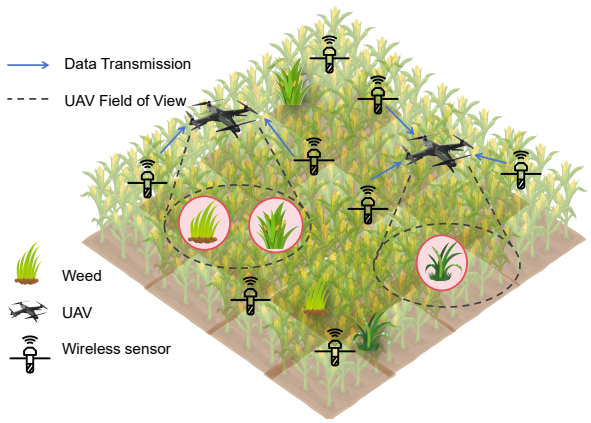


Fig. 1: Exemplary scenario of UAV-based smart farming, where UAVs perform weed detection, recognition, and data collection from wireless sensors simultaneously.

mounting lightweight cameras on UAVs, it is possible to detect the distribution of weeds and recognize the type of them, facilitating a smarter and greener weeding process.

Fig. 1 shows an exemplary scenario of smart farming, where n_{UAV} UAVs are used for weed detection and recognition in a farmland area. We assume that the UAVs have identical configurations and battery capacity BC, and they fly at constant altitude H. For the sake of convenience, we divide the farmland area into $N \times N$ fine-grained grids. The total flight duration of the UAVs is an episode λ, where the UAVs start from random locations. During each time step in an episode $t = 1, 2, \ldots, T_{\text{max}}$, we assume that the UAVs can detect the weeds that are within the FoV. It can recognize the density and types of weeds that are within the grid right below it using the embedded object detection algorithm. Meanwhile, there are n_{S} randomly distributed wireless sensors mounted on the ground, measuring real-time environmental data (e.g., sunlight, temperature, humidity, and soil pH). Apart from weed detection and recognition, the UAVs also collect data from wireless sensors that are within the communication range. The energy consumption for the UAVs should not exceed their battery capacities in an episode, and they will be fully charged before the start of the next episode.

B. System Model

1) Energy consumption model: The energy consumption of UAV$_i$ up to time t can be represented as:

$$E(i,t) = E_{\text{cmp}}(i,t) + E_{\text{cs}}(i,t) + E_{\text{fly}}(i,t), \tag{1}$$

where $E_{\text{cmp}}(i,t)$ denotes the cumulative computation energy consumption of UAV$_i$ at time t, $E_{\text{cs}}(i,t)$ denotes the cumulative communication and sensing energy consumption of UAV$_i$ at time t, and $E_{\text{fly}}(i,t)$ denotes the cumulative flight energy consumption of UAV$_i$ at time t.

The computational energy consumption $E_{\text{cmp}}(i,t)$ of UAV i at time t can be represented as:

$$E_{\text{cmp}}(i,t) = \int_0^t P_{\text{cmp}} \mathrm{d}t, \tag{2}$$

$$P_{\text{cmp}} = P_{\text{static}} + P_{\text{dynamic}}, \tag{3}$$

$$P_{\text{dynamic}} \approx C^* \cdot V^2 \cdot f^* \cdot \alpha^*, \tag{4}$$

where P_{cmp} represents computational power, P_{static} represents the static computing module power, P_{dynamic} represents the dynamic computing module power, C^* represents the load capacitance, V represents voltage, f^* represents clock frequency, and α^* represents the activity factor.

The communication and sensing energy consumption $E_{\text{cs}}(i,t)$ of UAV i at time t can be represented as:

$$E_{\text{cs}}(i,t) = \int_0^t P_{\text{cs}} \mathrm{d}t, \tag{5}$$

where P_{cs} represents the communication and sensing power.

The flight energy consumption $E_{\text{fly}}(i,t)$ of UAV i at time t can be represented as:

$$E_{\text{fly}}(i,t) = \int_0^t P_{\text{fly}}(i,t) \mathrm{d}t, \tag{6}$$

where $P_{\text{fly}}(i,t)$ denotes the flight power, calculated as:

$$P_{fly}(i,t) = c_1 |\boldsymbol{v}(i,t)|^2 + c_2 \frac{v_x^2(i,t) + v_y^2(i,t)}{|\boldsymbol{v}(i,t)|^3} + mg|\boldsymbol{v}(i,t)|, \tag{7}$$

where m_{UAV} denotes UAV mass, g is the gravitational acceleration, ρ_{air} denotes the air density, A_{UAV} is the UAV frontal area, η is the mechanical efficiency, and $\boldsymbol{v}(i,t)$ denotes the velocity vector. The components $v_x(i,t)$ and $v_y(i,t)$ represent the UAV's velocity in the x and y directions, respectively. c_1 and c_2 represent the drag and lift coefficients, respectively, which can be given by:

$$c_1 = \frac{1}{2} \rho \cdot A_{\text{UAV}} \cdot C_d, \tag{8}$$

$$c_2 = \frac{m_{\text{UAV}}^2}{\eta \cdot \rho_{\text{air}} \cdot n_{\text{prp}} \cdot \pi R_{\text{prp}}^2}, \tag{9}$$

$$A_{\text{UAV}} = A_{\text{surf}} + n_{\text{prp}} \pi R_{\text{prp}}^2, \tag{10}$$

where C_d denotes the air viscosity coefficient, n_{prp} denotes UAV propeller number, R_{prp} denotes the radius of the UAV propellers, and A_{surf} denotes the UAV surface area.

The remaining battery $B(i,t)$ can be presented as:

$$B(i,t) = BC - E(i,t), \tag{11}$$

where BC represents the battery capacity.

2) Communication model: Assuming that the packet loss rate of the transmission between UAV$_i$ and sensor$_j$ can be represented as:

$$\text{PLR}(i,j) = 1 - (1 - \text{BER}(i,j))^L, \tag{12}$$

where L represents the packet length, $\text{BER}(i,j)$ represents the bit error rate, with the binary phase shift keying (BPSK) encoding method in this paper, calculated as:

$$\text{BER}\,(i,j) = Q\left(\sqrt{2 \cdot \text{SINR}\,(i,j)}\right), \tag{13}$$

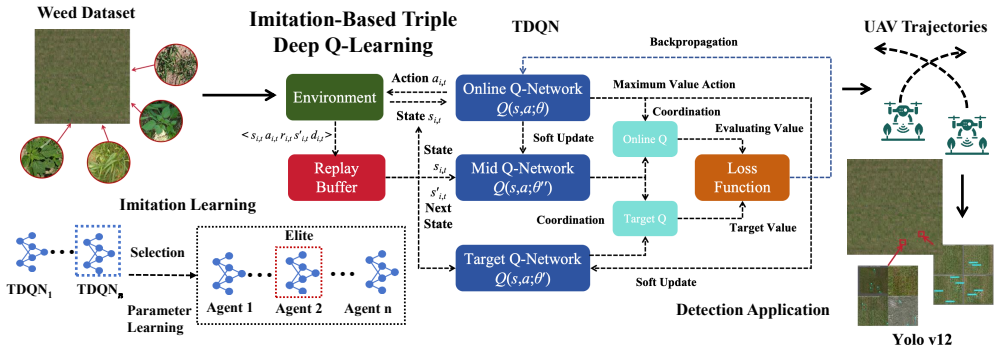


Fig. 2: The overview of our proposed ITDQN.

where $\mathrm{SINR}(i,j)$ represents the signal-to-interference-plus-noise ratio (SINR) between UAV_i and sensor_j, and $Q(x)$ means Q-function of the Gaussian distribution. They can be represented as:

$$\mathrm{SINR}(i,t) = \frac{P_{\mathrm{TX}} + G_t - \mathcal{PL}(i,j)}{N_{\mathrm{T}}}, \tag{14}$$

$$\mathcal{PL}(i,j) = 20\log_{10}\left(\frac{4\pi f_c Dis(i,j)}{c}\right) + 0.2 \cdot f_c^{0.3} \cdot Dis(i,j)^{0.6} \tag{15}$$

$$Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^{\infty} e^{-\frac{t^2}{2}} dt \approx \frac{1}{2} e^{-\frac{x^2}{2}}, \tag{16}$$

where P_{TX} represents sensor transmission power, $\mathcal{PL}(i,j)$ and $Dis(i,j)$ represent the path loss and distance between UAV_i and sensor_j, f_c represents carrier frequency, c represents light speed, and N_{T} represents the thermal noise, which can be represented as:

$$N_{\mathrm{T}} = k_B \times T_K \times Bw, \tag{17}$$

where k_B represents Boltzmann constant, T_K represents temperature in Kelvin, and Bw represents bandwidth.

The probability of successful data collection $p_{\mathrm{data}}(i,j)$ is represented as:

$$p_{\mathrm{data}}(i,j) = 1 - \mathrm{PLR}(i,j) \tag{18}$$

If the data is collected successfully, then the reward I_{data} is received.

C. Markov Decision Process

As the size of the farmland area scales up, the UAVs with limited FoV will only have a partial observation of the environment. Furthermore, due to the uncertain distribution of weeds and wireless sensors, optimization methods are not feasible for the problem. Therefore, we formulate the trajectory planning problem as the following MDP.

1) Agent: Each agent is a UAV that makes its decisions.

2) Environment: The environment is the farmland area.

3) States $s_{i,t}$: The states of UAV_i at time t include its location, the existence of weed within the FoV, the density of the weed, the location and the distance to the nearest wireless sensor that can be connected, the direction towards the highest density of wireless sensors that have been sampled, and the location and the distance to the nearest UAV. At each time t, the UAVs will broadcast their locations and the information of the grids they have recognized to other UAVs.

4) Actions $a_{i,t}$: The actions of UAV_i at time t are the flying directions, including North (N), Northeast (NE), East (E), Southeast (SE), South (S), Southwest (SW), West (W), and Northwest (NW).

5) Reward $r_{i,t}$: The reward UAV_i achieve at each time step t is denoted as:

$$r_{i,t} = P_{out} + P_{bat} + P_{clo} + I_{weed} + I_{data} + I_{exploit} + I_{explore} + b, \tag{19}$$

where P_{out}, P_{bat}, and P_{clo} are penalty terms for flying outside the regional boundary, running out of battery, and being too close to the neighboring UAV, respectively. I_{weed} and I_{data} are rewards for successfully recognizing weeds and collecting data, respectively. $I_{exploit}$ and $I_{explore}$ are incentives for exploitation (flying closer to the nearest sensor) and exploration (flying towards the direction of denser sensors), respectively, and b is a small constant to boost convergence.

IV. Imitation-Based Triple Deep Q-Network

A. Imitation-Based Reinforcement Learning

The architecture of our proposed ITDQN is depicted in Fig. 2. To enable agents to learn from high-performing individuals and lower the cost for exploration, we propose an *elite imitation mechanism* detailed below.

We introduce a mimicry cycle factor δ. During episodes determined by δ, each agent i does not update its policy $\pi_i(\theta)$ directly. Instead, it generates an action sequence $\mathcal{A}_i = \{a_{i,0}, \ldots, a_{i,n_{\mathrm{step}}}\}$ and receives the corresponding reward vector $\mathcal{R}_i = \{r_{i,0}, \ldots, r_{i,K_{\mathrm{end}}}\}$, where n_{step} represents the time step in an episode.

The elite (i.e., leading policy) is then identified based on the mean μ_i and variance σ^2 of the rewards in $\mathcal{R}_i$. The evaluation metric can be defined as:

$$E\mathcal{R}_i = \beta_1 \mu_i + \beta_2 \sigma_i^2, \tag{20}$$

Algorithm 1 Elite Imitation Mechanism.

Initialize policy network $\pi_i(\theta)$ for each agent i.
Initialize $\vartheta \leftarrow \vartheta^*$ and $\delta \leftarrow \delta^*$.
for episode $\lambda = 1, 2, \ldots, \lambda_{\max}$ **do**
  if $\lambda \mid \delta$ **then**
    for agent $i = 1, 2, \ldots, n_{\text{UAV}}$ **do**
      Generate $\mathcal{A}_i$ from $\pi_i(\theta)$.
      Obtain Reward $\mathcal{R}_i$ from the interaction between agent i and the environment.
    end for
    Compute the maximum reward evaluation parameter $E\mathcal{R}_i$ as Equation (20).
    Update $\pi_i(\theta)$ as Equation (21).
    Update parameters $\vartheta \leftarrow \alpha_1 \vartheta$ and $\delta \leftarrow \alpha_2 \delta$.
  end if
end for

where β_1 and β_2 are the weights for mean and variance for elite evaluation, respectively.

Other agents update their policy networks by soft-copying from the leading policy using a soft update parameter for elite ϑ. The update rule is formulated as:

$$\theta_i \leftarrow (1 - \vartheta)\theta_i + \vartheta\theta_i', \tag{21}$$

where θ_i' and θ_i denote the parameters of the elite and other agents' policy networks, respectively.

Furthermore, to maintain a certain level of exploration while benefiting from the imitation strategy, we gradually adjust the influence of the imitation process. Specifically, we gradually decrease ϑ and increase δ. We initialize the values of ϑ and δ as ϑ^* and δ^*. The detailed procedure of the elite imitation mechanism is shown in Algorithm 1.

B. Triple Deep Q-Learning

To accelerate learning and improve the accuracy and stability of Q-values, we introduce an additional Q-network, referred to as the *mediator Q-network (mid Q-network)*, over DDQN.

For each agent at each time step, the Q-network and the mid Q-network jointly generate Q-values $Q(s, a; \theta)$ and $Q''(s, a; \theta'')$ for action evaluation, based on which we construct a normal distribution function:

$$Q_{\text{online}}(s, a) \sim \mathcal{N}\left(\frac{Q(s, a; \theta) + Q''(s, a; \theta'')}{2}, \sigma^2\right), \tag{22}$$

where σ^2 indicates a fixed variance.

Subsequently, we sample a tuple $\langle S_i, A_i, R_i, S_i', D_i \rangle$ composed of $\langle s_{i,t},\ a_{i,t},\ r_{i,t},\ s_{i,t+1},\ d_{i,k} \rangle$ from the replay buffer $\mathcal{D}$, where $s_{i,t+1}$ represents the next state and $d_{i,k}$ represents whether task is done at the current time step.

Then the target value y_k of the k-th sample is expressed as:

$$y_k = r_{i,k} + (1 - d_k)\gamma Q_{\text{target}}(s_k', \arg\max_a Q_{\text{online}}(s_k', a)), \tag{23}$$

Algorithm 2 Triple Deep Q-Learning.

Initialize online $Q(s, a; \theta)$ network with weights θ.
Initialize target $Q'(s, a; \theta')$ network with weights $\theta' \leftarrow \theta$.
Initialize mid $Q''(s, a; \theta'')$ network with weights $\theta'' \leftarrow \theta$.
Initialize replay buffer $\mathcal{D}$.
for episode $\lambda = 1, 2, \ldots, \lambda_{\max}$ **do**
  for agent $i = 1, 2, \ldots, n_{\text{UAV}}$ **do**
    Update elite imitation strategy as Algorithm 1.
    for timestep $t = 1, 2, \ldots, T_{\max}$ **do**
      if With probability ε **then**
        Select a random action a.
      else
        Select $a = \arg\max_a Q_{\text{online}}(s, a)$ as (22).
      end if
      Execute action $a_{i,t}$, observe reward $r_{i,t}$, next state $s_{i,t}'$ and task done flag $d_{i,t}$.
      Store transition $(s_{i,t}, a_{i,t}, r_{i,t}, s_{i,t}', d_{i,t})$ into $\mathcal{D}$.
      Sample mini-batch $\{(s_j, a_j, r_j, s_j', d_j)\}$ from $\mathcal{D}$.
      Compute value target as (23).
      Perform the gradient descent step on loss as (25).
      Update target and mid networks as (26) and (27).
    end for
  end for
end for

where Q_{target} represents the evaluation for action value, which can be represented as:

$$Q_{\text{target}}(s, a) \sim \mathcal{N}\left(\frac{Q'(s, a; \theta') + Q''(s, a; \theta'')}{2}, \sigma^2\right). \tag{24}$$

The loss function $L(\theta)$ can be represented as:

$$L(\theta) = \frac{1}{B}\sum_{k=1}^{B}\left(y_k - Q(s_k, a_k; \theta)\right)^2, \tag{25}$$

where B represents the batch size.

The parameters θ' and θ'' are soft updated as:

$$\theta' \leftarrow \tau\theta + (1 - \tau)\theta', \tag{26}$$

$$\theta'' \leftarrow \tau\theta + (1 - \tau)\theta'', \tag{27}$$

where τ represents the soft update parameter for Q-networks.

The detailed procedure of triple deep Q-learning is shown in Algorithm 2.

V. Evaluation

A. Synthetic Simulation

1) Simulation Setup: In the synthetic simulation shown in Fig. 4a, there are 4 UAVs utilized for weed detection and recognition in a farmland area with 20×20 grids, where each grid represents an area of $20\ m \times 20\ m$. Meanwhile, they collect data from 40 wireless sensors randomly generated in the area. The farmland picture is concatenated using weed images from the Weed Image Detection Dataset [27], where each image is annotated with bounding boxes and weed categories for training, and YOLOv12 [26] is used for weed detection and recognition for each UAV. All measurements are averaged over 10 repeated samples.

TABLE I: Environmental parameters in the experiments.

Symbol	Definition	Value	Unit	Symbol	Definition	Value	Unit
$N \times N$	Size of farmland	20×20	grids	$n \times n$	Size of grid	20×20	m^2
FoV	UAV field of view	3×3	grids	d_{UAV}	UAV distance threshold	$20 \times \sqrt{2}$	m
n_{UAV}	Number of UAVs	4	-	n_{S}	Number of wireless sensors	40	-
H	UAV flying altitude	20	m	L	Packet length	20	Byte
BC	Battery capacity	51840	J	a_{max}	UAV maximum acceleration	$20 \times \sqrt{2}$	m/s^2
m_{UAV}	Mass of UAV	1	kg	f_c	Signal frequency	2.8	GHz
g	Gravitational acceleration	9.8	-	ρ_{air}	Air density	1.225	kg/m^3
v_{th}	Hovering speed threshold	0.1	m/s	C_d	Viscosity coefficient	0.5	–
n_{prp}	Propeller number	4	-	R_{prp}	Propeller radius	0.1	m
η	Mechanical efficiency	0.8	-	A_{surf}	UAV fuselage area	0.01	m^2
T_{max}	Length of an episode	200	s	P_{static}	Static power	4	W
t	Time step in an episode	1	s	k_B	Boltzmann constant	1.38×10^{-23}	-
T_K	Temperature in Kelvin	298	K	Bw	Bandwidth	20	MHz
P_{cs}	Communication and sensing power	30	dBm	P_{TX}	Sensor transmission power	17	dBm
α^*	Activity factor	0.5	-	C^*	Load capacitance	6.4	nF
c	Speed of light	3×10^8	m/s	f^*	Clock frequency	200×10^6	Hz
V	Voltage	5	V				

TABLE II: Hyperparameters in the experiments.

Symbol	Definition	Value	Symbol	Definition	Value
P_{out}	Penalty for flying out of farmland	10	P_{bat}	Penalty for battery outage	10
P_{clo}	Penalty for collision avoidance	$0.1 \times \Delta d$	b	Constant to boost convergence	0.1
I_{weed}	Reward for weed recognition	2	I_{data}	Reward for data collection	2
$I_{exploit}$	Incentive for exploitation	0.05	$I_{explore}$	Incentive for exploration	0.1
γ	Discounted factor	0.99	M	Memory size	2^{16}
r	Learning rate	10^{-4}	D	Replay buffer size	2^{16}
B	RL training batch size	128	ϵ	Initial exploration probability	1
ϵ_{min}	Minimum exploration probability	0.01	ϵ_{decay}	Exploration probability decay rate	0.995
λ_{max}	Maximum training episode	1000	hid	Hidden dimension	256
ϑ^*	Initial elite soft update parameter	0.1	δ^*	Initial mimicry cycle factor	10
α_1	Elite soft update parameter decay rate	$1/2$	α_2	Mimicry cycle factor increase rate	2
β_1	Weight of mean for elite evaluation	1	β_2	Weight of variance for elite evaluation	0.01
τ	Soft update parameter for Q-network	0.01	σ^2	Fixed variance for Q-network	0.01

TABLE III: Performance metrics in the experiments (*indicates the usage under the assumption of a pre-known environment, leading to the upper limit of the weed detection rate, but are infeasible in real-world environments).

Algorithm	Energy consumption (J)	Weed recognition rate	Data collection rate	Task completion time (s)	Inference time (ms)
Synthetic simulation					
ACO*	4435.1718	97.50%	66.75%	41.00	114.71
PSO*	4867.8715	97.50%	83.50%	45.00	130.90
GA*	5300.5711	97.50%	73.75%	49.00	138.09
DQN	4220.9571	75.74%	91.40%	34.53	5.79
DDQN	4669.3422	75.00%	91.11%	38.80	6.16
ITDQN	5608.8414	79.43%	98.05%	47.45	6.53
Real-world demonstration					
ITDQN	31132.5696	85.08%	79.75%	91.75	3.45

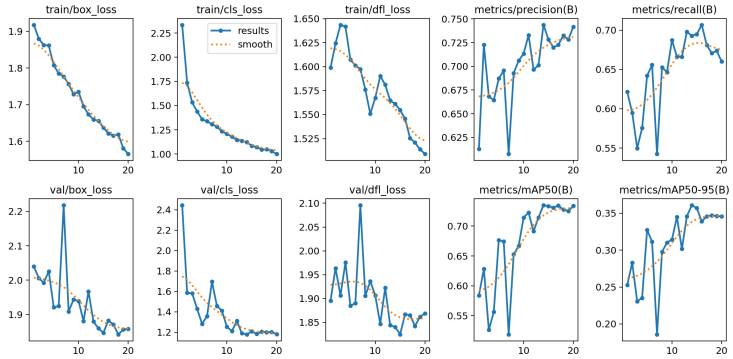
2) Weed Detection and Comparison: Fig. 3a shows the weed detection and recognition results in six exemplary grids with the weed type "ridderzuring". It can be seen that most of the weeds are successfully detected by the bounding boxes and categorized with correct labels. The inference time using YOLOv12 is 2.73 milliseconds. Fig. 3b further details the performance metrics, which yield precision, recall, F1-score, and IoU values above 70%.

3) Comparison with Baselines: In the comparative study, we choose three heuristics for comparison, namely ant colony optimization (ACO) [28], particle swarm optimization (PSO) [29], and genetic algorithm (GA) [30]. As MARL baselines, we choose DQN [14] and DDQN [15].

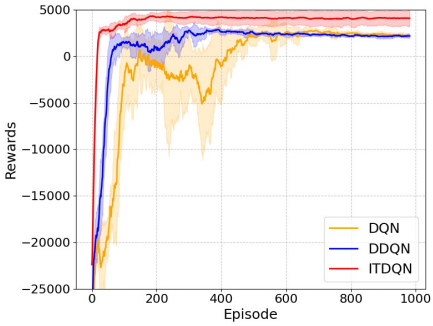
Fig. 3c compares the reward versus the number of episodes of DQN, DDQN, and ITDQN. Among these algorithms, DQN has the slowest convergence speed and lowest stability. In contrast, DDQN addresses the overestimation bias by duplicating the DQN architecture and redefining the Q-learning update equation. Thus, it achieves comparable reward with DQN, but with significantly higher speed and less fluctuation. Moreover, our proposed ITDQN achieves the highest convergence speed because the novel elite imitation mechanism enables the rapid learning and refinement of policies for multiple UAVs. Furthermore, the improved triple Q-network architecture facilitates the coordination between online and target Q-networks, surpassing existing algorithms in performance and stability. Therefore, we can conclude that our proposed ITDQN is superior to DQN and DDQN methods in terms of performance, efficiency, and stability.

Performance metrics obtained by various algorithms are listed in Table III. Compared with heuristics such as ACO, PSO, and GA, our proposed ITDQN achieves superior perfor-

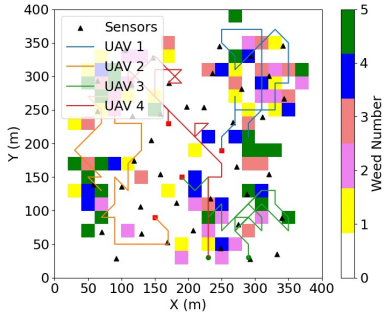
(a) Weed detection and recognition results.



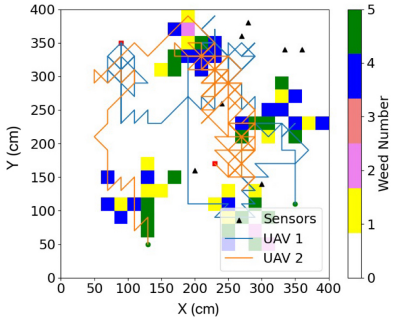
(b) Weed detection and recognition metrics using YOLOv12 [26].



(c) Comparison of reward versus number of episodes using DQN, DDQN, and ITDQN.



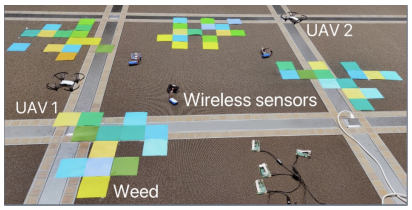
(d) UAV trajectories in synthetic simulation, where the weed density is derived from the weed recognition results shown in (3a).



(e) UAV trajectories in real-world demo, where the weed density is represented by markers with different colors shown in (4b).

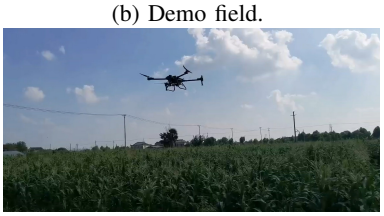
Fig. 3: Experimental results in simulated and real-world environments, following our three-step methodology in Fig. 4.

(a) Simulation field.



(b) Demo field.

(c) Demo equipment.



(d) Real-world field.

Fig. 4: Our evaluation follows a three-step methodology as follows. Step 1: Simulation in a synthetic field (a), step 2: Demonstration in an indoor area (b and c), and step 3: Evaluation in a real-world field as our future work (d).

mance in data collection and inferior performance in weed recognition. The heuristics are used under the assumption of pre-known environments. However, due to the uncertain distribution of weeds and the random starting points of UAVs, the heuristics are infeasible to deploy in real-world environments. In addition, compared with MARL algorithms, heuristics have much higher inference times. The heuristics rely on complex rule-based logic or search procedures, which are computationally intensive. In contrast, MARL policies, once trained, are typically executed through efficient neural network inference, allowing for scalable and real-time decision-making.

Compared to DQN and DDQN, which exhibit similar performance, our proposed ITDQN has higher energy consumption and task completion time, but is still within the battery capacity and episode length requirements. The inference time of DQN is lower than that of DDQN, and both are lower than ITDQN, as they use 1, 2, and 3 layers of Q-networks, respectively. Most importantly, ITDQN achieves the highest task performance, surpassing DDQN by 4.43% in weed recognition rate and 6.94% in data collection rate.

4) UAV Trajectories: The visualization of UAV trajectories in the simulation is depicted in Fig. 3d. The UAV trajectories are aligned with the weed distribution shown in the heatmap. Furthermore, the four UAVs collaborate well for weed detection, recognition, and data collection, with each of them responsible for a cluster of weeds and a data collection region.

B. Real-World Demonstration

In the real-world demonstration, we utilize 2 DJI Tello UAVs flying in a $4\ m \times 4\ m$ indoor area (cf. Fig. 4b). We use different colored markers to represent weeds of varying densities, and use Internet of Things (IoT) devices and Raspberry

Pis to represent 8 wireless sensors (cf. Fig. 4c). Table III shows that our proposed ITDQN consistently performs well, with an 85.05% weed recognition rate and a 79.75% data collection rate. Furthermore, Fig. 3e shows that compared to that in simulation, the UAV trajectories in the real-world demonstration encounter more zig-zag detours, indicating more challenging conditions when UAVs explore in the real-world environments.

VI. Conclusion

We formulate the trajectory planning for UAV-based smart farming as an MDP and employ MARL. We introduce a novel ITDQN algorithm, which integrates an elite imitation mechanism to lower exploration costs and incorporates a mediator Q-network over a conventional DDQN to enhance performance, training efficiency, and stability. Experimental results in simulated and real-world environments show that ITDQN surpasses state-of-the-art baselines. We will evaluate the effectiveness of ITDQN in real farmland environments.

References

[1] S. Wolfert, L. Ge, C. Verdouw, and M.-J. Bogaardt, "Big data in smart farming – a review," *Agricultural Systems*, vol. 153, pp. 69–80, 2017. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0308521X16303754

[2] V. Moysiadis, P. Sarigiannidis, V. Vitsas, and A. Khelifi, "Smart farming in europe," *Computer Science Review*, vol. 39, p. 100345, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1574013720304457

[3] G. Idoje, T. Dagiuklas, and M. Iqbal, "Survey for smart farming technologies: Challenges and issues," *Computers & Electrical Engineering*, vol. 92, p. 107104, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0045790621001117

[4] N. Islam, M. M. Rashid, F. Pasandideh, B. Ray, S. Moore, and R. Kadel, "A review of applications and communication technologies for internet of things (iot) and unmanned aerial vehicle (uav) based sustainable smart farming," *Sustainability*, vol. 13, no. 4, 2021. [Online]. Available: https://www.mdpi.com/2071-1050/13/4/1821

[5] P. Parameswari, V. Sujitha, J. Surya, B. A. Kumar, and S. Aakash, "Optimized uav trajectory planning for precision agriculture using wireless sensor networks," in *2025 3rd International Conference on Advancements in Electrical, Electronics, Communication, Computing and Automation (ICAECA)*, 2025, pp. 1–6.

[6] J. Choton and W. Hsu, "Coverage path planning in precision agriculture: Algorithms, applications, and key benefits," 12 2024.

[7] R. I. Mukhamediev, K. Yakunin, M. Aubakirov, I. Assanov, Y. Kuchin, A. Symagulov, V. Levashenko, E. Zaitseva, D. Sokolov, and Y. Amirgaliyev, "Coverage path planning optimization of heterogeneous uavs group for precision agriculture," *IEEE Access*, vol. 11, pp. 5789–5803, 2023.

[8] X. Liu, G. Li, H. Yang, N. Zhang, L. Wang, and P. Shao, "Agricultural uav trajectory planning by incorporating multi-mechanism improved grey wolf optimization algorithm," *Expert Systems with Applications*, vol. 233, p. 120946, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417423014483

[9] R. van Essen, E. van Henten, L. Kooistra, and G. Kootstra, "Adaptive path planning for efficient object search by uavs in agricultural fields," *Smart Agricultural Technology*, vol. 12, p. 101075, 2025. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2772375525003089

[10] H. Fu, Z. Li, W. Zhang, Y. Feng, L. Zhu, X. Fang, and J. Li, "Research on path planning of agricultural uav based on improved deep reinforcement learning," *Agronomy*, vol. 14, no. 11, 2024. [Online]. Available: https://www.mdpi.com/2073-4395/14/11/2669

[11] G. Zhang, J. Liu, W. Luo, Y. Zhao, R. Tang, K. Mei, and P. Wang, "A shortest distance priority uav path planning algorithm for precision agriculture," *Sensors*, vol. 24, no. 23, 2024. [Online]. Available: https://www.mdpi.com/1424-8220/24/23/7514

[12] C. J. C. H. Watkins and P. Dayan, "Q-learning," *Machine Learning*, vol. 8, no. 3, pp. 279–292, 1992.

[13] J. Tsitsiklis and B. Van Roy, "An analysis of temporal-difference learning with function approximation," *IEEE Transactions on Automatic Control*, vol. 42, no. 5, pp. 674–690, 1997.

[14] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. A. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, pp. 529–533, 2015.

[15] H. van Hasselt, A. Guez, and D. Silver, "Deep reinforcement learning with double q-learning," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 30, no. 1, Mar. 2016. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/10295

[16] Z. Wang, T. Schaul, M. Hessel, H. Hasselt, M. Lanctot, and N. Freitas, "Dueling network architectures for deep reinforcement learning," in *Proceedings of The 33rd International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, M. F. Balcan and K. Q. Weinberger, Eds., vol. 48. New York, New York, USA: PMLR, 20–22 Jun 2016, pp. 1995–2003. [Online]. Available: https://proceedings.mlr.press/v48/wangf16.html

[17] T. Schaul, J. Quan, I. Antonoglou, and D. Silver, "Prioritized experience replay," *CoRR*, vol. abs/1511.05952, 2015. [Online]. Available: https://api.semanticscholar.org/CorpusID:13022595

[18] M. Fortunato, M. G. Azar, B. Piot, J. Menick, I. Osband, A. Graves, V. Mnih, R. Munos, D. Hassabis, O. Pietquin, C. Blundell, and S. Legg, "Noisy networks for exploration," *CoRR*, vol. abs/1706.10295, 2017. [Online]. Available: http://arxiv.org/abs/1706.10295

[19] M. G. Bellemare, W. Dabney, and R. Munos, "A distributional perspective on reinforcement learning," in *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ser. ICML'17. JMLR.org, 2017, p. 449–458.

[20] W. Dabney, M. Rowland, M. G. Bellemare, and R. Munos, "Distributional reinforcement learning with quantile regression," in *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*, ser. AAAI'18/IAAI'18/EAAI'18. AAAI Press, 2018.

[21] W. Dabney, G. Ostrovski, D. Silver, and R. Munos, "Implicit quantile networks for distributional reinforcement learning," in *Proceedings of the 35th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, J. Dy and A. Krause, Eds., vol. 80. PMLR, 10–15 Jul 2018, pp. 1096–1105. [Online]. Available: https://proceedings.mlr.press/v80/dabney18a.html

[22] M. Hessel, J. Modayil, H. van Hasselt, T. Schaul, G. Ostrovski, W. Dabney, D. Horgan, B. Piot, M. Azar, and D. Silver, "Rainbow: combining improvements in deep reinforcement learning," in *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*, ser. AAAI'18/IAAI'18/EAAI'18. AAAI Press, 2018.

[23] A. Ly, R. Dazeley, P. Vamplew, F. Cruz, and S. Aryal, "Elastic step dqn: A novel multi-step algorithm to alleviate overestimation in deep qnetworks," 2022. [Online]. Available: https://arxiv.org/abs/2210.03325

[24] M. J. Hausknecht and P. Stone, "Deep recurrent q-learning for partially observable mdps," *CoRR*, vol. abs/1507.06527, 2015. [Online]. Available: http://arxiv.org/abs/1507.06527

[25] P. Lottes, R. Khanna, J. Pfeifer, R. Siegwart, and C. Stachniss, "Uav-based crop and weed classification for smart farming," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, 2017, pp. 3024–3031.

[26] Y. Tian, Q. Ye, and D. Doermann, "Yolov12: Attention-centric real-time object detectors," 2025. [Online]. Available: https://arxiv.org/abs/2502.12524

[27] J. Dalmotra, "Weed detection," 2023. [Online]. Available: https://www.kaggle.com/dsv/6675836

[28] M. Dorigo, M. Birattari, and T. Stutzle, "Ant colony optimization," *IEEE Computational Intelligence Magazine*, vol. 1, no. 4, pp. 28–39, 2006.

[29] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of ICNN'95 - International Conference on Neural Networks*, vol. 4, 1995, pp. 1942–1948 vol.4.

[30] S. Mirjalili, "Genetic algorithm," *Handbook of Machine Learning*, 2018. [Online]. Available: https://api.semanticscholar.org/CorpusID:17088312