Computers, Materials & Continua

Tech Science Press

https://doi.org/10.32604/cmc.2026.080008

ARTICLE

IRL-TP: Deep Inverse Reinforcement Learning-Based Trajectory Planning for UAVs in Complex and Interference-Constrained Environments

Xuan-Thuc Nguyen[1], Le-Minh Nguyen[1], Ngoc-Quynh Nguyen[1], Nhu-Nghia Bui[2], Dinh-Quy Vu[3,*] and Thai-Viet Dang[2,*]

[1]Viettel High Technology Industries Corporation–Viettel Group, Hanoi, Vietnam
[2]Department of Mechatronics, School of Mechanical Engineering, Hanoi University of Science and Technology, Hanoi, Vietnam
[3]Department of Vehicle and Energy Conversion Engineering, School of Mechanical Engineering, Hanoi University of Science and Technology, Hanoi, Vietnam

*Corresponding Authors: Dinh-Quy Vu. Email: quy.vudinh@hust.edu.vn; Thai-Viet Dang. Email: viet.dangthai@hust.edu.vn

Received: 01 February 2026; Accepted: 14 April 2026; Published: 15 June 2026

ABSTRACT: The development of unmanned automated vehicles (UAVs) has become a key focus in aerial robotics, fueling the need for navigation systems capable of performing complex and delicate tasks with speed and precision. However, the end-to-end path tracking process often encounters challenges in learning efficiency, generalization, and varying environmental conditions. In this paper, we propose the novel IRL-TP framework for learning-based UAVs' trajectory planning that employs a deep inverse reinforcement learning (IRL) approach. Firstly, the RL-based path planner must develop a reward function that effectively captures flight safety, collision avoidance, trajectory smoothness, and navigation efficiency within constrained environments filled with numerous obstacles. To achieve optimal results, a deep reward network is constructed to parametrize the unknown reward function, which effectively and implicitly models the satisfaction of multiple objectives. The regularization of entropy through the learned reward function is utilized to optimize the continuous control policy and improve stability and exploration ability during training with a soft actor-critic (SAC) agent. By combining the reward function inference and policy learning processes, the proposed framework empowers the UAVs to mimic expert behavior and create highly generalized navigation strategies in the "potential map". In experimental environments with a dense obstacle level, our method achieves a success rate of 97.6% while maintaining an instability metric as low as 0.044 throughout the process. Furthermore, the number of episodes needed to converge the parameters was much faster than other methods (~340). The proposed model not only achieves rapid convergence and a reward value 1.6 times higher in the first 200 training episodes and 1.3 times higher after the entire training process, but also demonstrates an impressive inference time of 2.6 ms per step compared to the basic IRL framework. Compared to state-of-the-art methods—including DQN, PPO, SAC, BC, and GAIL—our approach achieves superior trajectory efficiency, enhanced safety margins, smoother motion, and greater training stability, even in complex 3D environments.

KEYWORDS: Interference-constrained environments; inverse reinforcement learning; unmanned automated vehicles (UAV); soft actor-critic (SAC); trajectory planning

1 Introduction

The rapid advancement of UAVs has catalyzed a paradigm shift across various control domains, including autonomous infrastructure inspection, intelligent search-and-rescue, and sophisticated industrial automation systems [1]. Deep learning (DL) has emerged as a fundamental cornerstone for high-dimensional

sensory data processing, enabling UAVs to interpret multifaceted environmental data with unprecedented accuracy. Building upon this, reinforcement learning (RL) [2] has gained significant traction by providing a robust framework for agents to derive optimal control policies through continuous agent-environment interactions within Markov decision processes (MDPs) [3]. At its core, RL algorithms seek to optimize decision policies by maximizing cumulative rewards, yet the design of precise reward functions remains a critical bottleneck. In complex dynamic environments, manually crafting a reward function that encompasses all nuances of safety, energy efficiency, and mission objectives is often insurmountable. To bridge this gap, researchers have pioneered IRL to automate the recovery of latent reward structures from observed expert demonstrations [4]. By extracting the underlying intent and cost functions of an expert, IRL offers a more adaptive and resilient approach to decision-making in autonomous systems. The transition from fixed reward engineering to demonstration-based reward recovery is particularly vital for UAVs' trajectory planning. IRL enables aerial platforms to mimic human-like decision-making processes, effectively capturing implicit trade-offs inherent in expert trajectories without manual reward shaping. Recent methodologies have successfully harnessed deep neural networks as function approximators to handle the continuous and high-dimensional state spaces characteristic of aerial robotics [5]. This integration allows for the approximation of arbitrary nonlinear reward functions, significantly enhancing the empirical performance of autonomous agents. Consequently, IRL stands as a promising frontier for developing adaptive trajectory planning systems that can generalize across diverse and high-stakes operational scenarios.

The field of trajectory planning has progressively transitioned from traditional heuristic approaches to sophisticated data-driven techniques, with RL gaining widespread application in autonomous vehicles, strategic gaming, and robotic manipulation. However, traditional RL methodologies require manual specification of cost functions of reward shaping to inconsistent or suboptimal optimization outcomes across diverse tasks [6]. In the context of UAV navigation, inadequately defined cost functions can significantly hinder convergence rates and optimization performance, particularly in dynamic, obstacle-rich environments where multi-objective trade-offs—such as safety, energy efficiency, and smoothness—are challenging to quantify [7]. To address these issues, IRL frameworks have been developed to infer unknown cost functions directly from observed expert behaviors, thereby effectively capturing the underlying intent characteristic of human-level decision-making. Early IRL methods [8], including margin-based apprenticeship learning and maximum margin planning, established the foundation for matching feature expectations but were substantially limited by the problem of reward ambiguity. This ambiguity arises when multiple distinct reward functions can equally explain the same expert policy, complicating strategy optimization in complex three-dimensional terrains. The Maximum Entropy IRL (MaxEnt-IRL) framework [9] systematically addressed this issue by maximizing the log-likelihood of demonstrations while preserving policy entropy to resolve reward symmetry. Subsequent advancements extended this framework to incorporate causal conditional probability distributions, enabling temporal modeling of sequentially revealed information—a critical capability for UAVs operating under dynamic sensory constraints. To manage the high-dimensional state spaces characteristic of aerial robotics, Deep Inverse Reinforcement Learning (DIRL) [10] was introduced, employing deep neural networks to approximate nonlinear reward functions in an end-to-end fashion. Concurrently, sampling-based maximum entropy inverse optimal control (ME-IOC) methods [11] were developed to overcome the computational intractability associated with normalizing constants during reward recovery. The field further progressed with adversarial frameworks such as Generative Adversarial Imitation Learning (GAIL) [12] and Adversarial Inverse Reinforcement Learning (AIRL) [13], which utilize adversarial training to derive robust and generalizable reward functions. Nevertheless, these adversarial approaches often exhibit high sample complexity and training instability when applied to continuous UAV control tasks. Recent research has sought to enhance AIRL through active learning and adaptive

algorithms that incorporate mixed expert demonstrations, thereby reducing annotation costs. Despite these advancements, specific challenges remain in UAV trajectory planning. For example, integration with control theory via Model Predictive Control (MPC) [14] and Differential Dynamic Programming (DDP) [15] offer real-time computational efficiency but typically depends on imitation-based loss functions that assume expert data originates from open-loop control. This assumption fails to capture the feedback dynamics inherent in expert demonstrations, which are essential for reactive navigation in unstructured environments. Reconceptualizing IRL from a closed-loop perspective has led to the development of DDP-based gradient solvers that more accurately model this feedback nature. Moreover, while the complexity of nonlinear systems can be addressed through Takagi-Sugeno fuzzy models to guarantee asymptotic stability, and robust trajectory tracking quality in large-scale, complex settings [16]. In multi-agent IRL (MA-IRL) scenarios [17], methodologies such as graph attention mean field theory (GAMF) and theory of mind (ToM) have been proposed to model heterogeneous interactions; however, these approaches frequently demand substantial computational resources, limiting their applicability on Edge AI platforms. Additionally, the challenge of sparse rewards—where UAVs receive feedback only upon task completion—impedes efficient learning due to the infrequent occurrence of informative states. These limitations highlight the necessity for a framework that harmonizes traditional reward maximization with attributes such as optimal navigation policy and system stability. This need motivates the development of the proposed IRL-TP framework, which employs a deep reward network in conjunction with entropy-regularized actor-critic learning to autonomously infer latent reward functions. By integrating reward inference with stable continuous policy optimization, IRL-TP facilitates feasible path planning in complex, interference-constrained environments while maintaining the computational efficiency required for real-time UAV operations.

Driven by the need for more sample-efficient and robust navigation in interference-constrained environments, the paper proposes a novel IRL-TP framework specifically engineered for UAVs' navigation in constrained environments. A process of modeling the environment thru a hexagonal grid mechanism to limit the scope and focus on effectively exploiting the attributes of expert actions, serving as an optimal reference for predicting UAV navigation in real-world environments. The goal is to leverage the generality of IRL over traditional RL, while easily coordinating the flow of behavioral observations into specific behavioral trajectories. The ambiguity and difficulty in replicating the evaluation of action effectiveness through rewards are resolved. Unlike the traditional reinforcement learning approaches that heavily depend on manual engineering of reward functions, the proposed method infers a latent reward function directly from expert demonstrations, which allows for automatic discovery of implicit navigation objectives such as collision avoidance, motion smoothness, and efficient maneuvering in constrained spaces. The central aspect of this work is the integration of a deep reward network within maximum entropy IRL formulation with SAC agent as policy optimizer. By combining these features, the framework can recover a highly expressive nonlinear reward function and learn to maintain definite continuous control policies in high-dimensional state-action spaces. Unlike previous IRL techniques that rely on classical or discrete policy optimization schemes, the new framework employs entropy-regularized actor-critic learning to aid exploration, improve training robustness, and extend generalization capability to invisible confined environments. Consequently, the proposed technique not only replicates expert behaviors but also empowers the UAVs to acquire more effective and adaptive navigation strategies, while eliminating the need for manual reward shaping, which remains a significant bottleneck in practical use of autonomous UAVs. Finally, practical contributions include a verifiable navigation system that demonstrates superior resilience in diverse, high-noise operational theaters compared to traditional reinforcement learning baselines.

The main contributions of this work are as follows:

- Propose the deep IRL framework for the UAV navigation problem in environments with high obstacle density, where the latent reward function is automatically deduced from expert demonstrations instead of being manually constructed.
- Combine expert-based policy initialization with the SAC framework for enabling both nonlinear reward recovery and stable continuous policy optimization under entropy regularization.
- Test and ensure the superior UAV's navigation performance in comparison with the state-of-the art methods like RL/IL-based optimization and heuristic approaches.

The remainder of this paper is organized as follows: Section 2 reviews related work in the field. Section 3 details the architecture of proposed inverse reinforcement learning-based trajectory planner. Section 4 presents experimental results and comparative analyses. Finally, Section 5 offers concluding remarks and outlines directions for future research.

2 Related Works

2.1 Heuristic Algorithms

Classical path planning for UAVs has traditionally relied on search-based methods such as A* and Dijkstra's algorithm [18], or sampling-based approaches like rapidly exploring random trees (RRT) [19], and probabilistic roadmaps (PRM) [20]. While these algorithms are effective in structured environments, they often encounter the "dimensionality curse" and increased computational overhead when navigating unknown or expansive 3D spaces [21]. Furthermore, traditional motion planners struggle with dense obstacle distributions and can become trapped in local optima. To address these limitations, recent research has shifted toward metaheuristic algorithms inspired by biological mechanisms. These include particle swarm optimization (PSO) [22], grey wolf optimizer (GWO) [23], and whale optimization algorithm (WOA) [24], which offer enhanced search capabilities in cluttered environments. A notable advancement is the improved crested porcupine optimizer (ICPO) [25], which utilizes a visuo-auditory synergy perspective and periodic retreat strategies to balance exploration and exploitation, effectively avoiding local optima in complex mountainous and urban terrains. Despite their speed and efficiency, these heuristic methods often require extensive parameter tuning and may lack the real-time adaptability needed for highly dynamic interference-constrained environments.

2.2 Reinforcement Learning Algorithms

Reinforcement Learning has emerged as a transformative paradigm for map-free navigation, allowing UAVs to learn optimal policies through direct environmental interaction. Early applications primarily focused on discrete action spaces using deep Q-networks (DQN) [26], but the demand for smooth flight dynamics led to the adoption of continuous control frameworks like deep deterministic policy gradient (DDPG) and twin delayed DDPG (TD3) [27]. To address the challenges of partially observable Markov decision processes (POMDPs) [28], researchers have integrated long short-term memory (LSTM) [29] or recurrent neural networks (RNNs) [30] into DRL architectures, providing the agent with a temporal memory to capture latent state information and avoid "local traps". Advanced variants such as proximal policy optimization (PPO) [31] and SAC [32] have further improved training stability and exploration efficiency in unknown 3D spaces. Moreover, MARDPG [33] utilize centralized training and decentralized execution to coordinate multiple UAVs, ensuring safe navigation without requiring constant inter-agent communication. Despite these gains, purely DRL-based methods often suffer from sparse reward signals and trajectories that lack the kinematic smoothness required for energy-efficient operations. The adverse characteristics

dependent on the forms of each type of reinforcement learning model carried over to the navigation problem are also a significant challenge. The ability to deploy in real-world complex environments, which is quite different from the limited training scenarios, significantly affects the stability of UAVs.

2.3 Inverse Reinforcement Learning

The inherent difficulty of manually defining complex reward functions in interference-constrained environments has catalyzed the development of inverse reinforcement learning (IRL), which recovers reward structures directly from expert demonstrations. In spatial search and exploration, the adaptive submodular inverse reinforcement learning (ASIRL) algorithm demonstrates high efficacy by learning reward functions in the Fourier domain and utilizing compressed sensing for spatial recovery. To tackle scenarios where expert data is imperfect or environmental interference is high, hybrid learning frameworks have been introduced. For example, imitation augmented deep reinforcement learning (IADRL) [34] facilitates cooperative UAV-UGV coalitions to overcome power and payload limitations. Another critical hybridization involves hierarchical architectures employing an upper-layer learning module (DRL/IRL) for high-level local target generation and a lower-layer classical motion planner to ensure the resulting paths are smooth, collision-free, and dynamically feasible [35]. These systems often utilize event-triggered mechanisms to optimize computational efficiency, activating decision-making only when environmental changes or obstacles are detected. Furthermore, E-GAIL [36] provides an effective policy with high efficiency to improve long-term rewards of GAIL and incorporating negative actions into generated trajectories. By combining the cognitive-level decision-making of IRL with the physical-level precision of traditional controllers, these hybrid approaches provide the necessary resilience for UAV navigation in highly complex and interference-prone environments. Generalization of training activities and the exploitation of specialized attributes in actions are emphasized. This is a prerequisite factor in the design of strategic planning problems to navigate effectively in various environmental scenarios.

3 Proposed Method

3.1 Expert and Features

A three-dimensional (3D) hexagonal grid is used to model the UAVs' environment within a constrained space, enabling the IRL-TP's framework (see Fig. 1). The hexagonal cells are arranged in a horizontal planar layout and stacked vertically along the altitude axis, creating a volumetric flight space. Each cell represents a discrete, localized region of physical significance, as illustrated in Fig. 2. The UAV's continuous spatial position is discretized by mapping it to the corresponding grid cell, which then defines its state representation.

$$s_t = Hex\left(p_t\right) \tag{1}$$

where $p_t = (x_t, y_t, z_t)$ represents the continuous three-dimensional position of the UAV at time step t. The $Hex(\cdot)$ function represents a spatial mapping from the continuous domain to a discrete hexagonal grid. Specifically, it assigns the continuous position pt to the corresponding hexagonal cell based on the spatial partitioning. To generate the feature vector $\phi(s_t)$ used as input for the deep reward network, each state is described by environmental attributes such as an obstacle indicator, the distance to the nearest obstacle, and a risk level. This description is achieved by using hexagonal meshes, which effectively maintain geometric symmetry, minimize directional bias in movement, and enable real-time estimation of state visitation frequencies and expected values, as required in IRL-TP model. As a result, this method allows the model to grasp the underlying spatial structure.

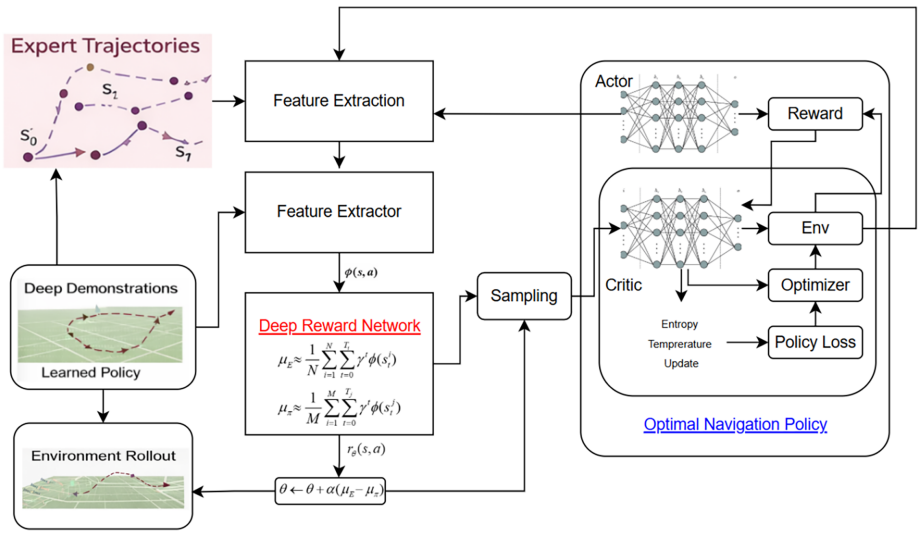

Figure 1: Proposed IRL-TP's architecture for UAVs' trajectory planning.

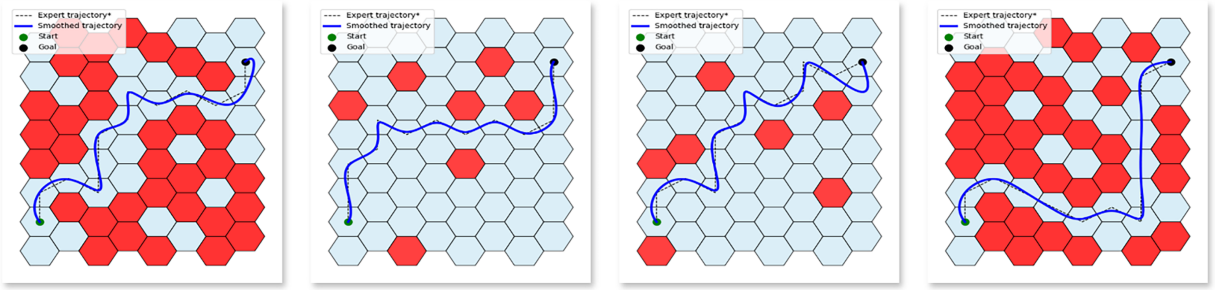

Figure 2: The cross-section of the maps models the environment in hexadecimal blocks. Expert orbital lines have been created with red blocks are high-risk zones and light blue blocks are potential zones.

3.2 Deep Reward Network

The implicit guidelines for achieving expert-level optimal behavior in constrained environments include avoiding bottlenecks, prioritizing wide corridors and other obstacles, reducing speed when entering high-risk zones, and maintaining a safe distance from boundaries. Translating these guidelines into practical reward structures poses significant challenges. A "potential map" within the state space illustrates how deep reward networks can capture behavioral patterns as nonlinear functions. By utilizing SAC, the agent is guided to develop stable control policies that generalize effectively to new environmental conditions through the learned reward function. Consequently, the task of navigating UAVs in confined spaces is modeled as an MDP.

$$M = [S, A, P, r, \gamma] \tag{2}$$

where S represents the state space of the UAV, including variables such as position, velocity, obstacle proximity, and corridor orientation. The set of possible actions available in each state is denoted by A. Environmental factors are captured by P. The reward function, which is unknown, is represented by r, while γ denotes the discount factor used throughout the process. The reward function is approximated using a parameterized deep neural network, as expressed in the following Eq. (3):

$$r_\theta(s,a) = f_\theta(\phi(s), a) \tag{3}$$

where the function $\phi(s)$ acts as a characteristic mapping of the state, capturing essential geometric and safety-related information within constrained environments. This includes parameters such as the distance to the nearest obstacle, proximity to corridor boundaries, obstacle density, target orientation, and the UAV's kinematic properties. Vector a represents continuous control action, such as velocity or axial acceleration. The reward network architecture is built on a fully connected multilayer perceptron (MLP) with L layers, leveraging the fundamental principles of these networks. Two hidden layers use the ReLU activation function, which enhances the model's ability to represent nonlinear relationships. Additionally, the output layer employs a linear activation function, providing the necessary flexibility to represent reward values.

$$h^{(l+1)} = \sigma\left(W^{(l)}h^{(l)} + b^{(l)}\right), \quad l = 1, \ldots, L-1 \tag{4}$$

Combine Eqs. (3) with (4) to obtain the following Eq. (5):

$$r_\theta(s,a) = W^{(L)}h^{(L)} + b^{(L)} \tag{5}$$

Next, integrating actions into the reward network allows the model to learn complex criteria, such as identifying which actions are safe within a given spatial context, rather than simply evaluating static states. Unlike open areas, confined spaces demand features that accurately represent the geometric configuration of the surrounding environment. The feature vector $\phi(s_t)$ was used to calculate several metrics, including the minimum distance to obstacles in various directions, the distance from the center of the flight corridor, the angular deviation between the UAV's current heading and the target direction, the remaining distance to the target, as well as the UAV's current velocity and acceleration. By leveraging these features, the reward network can abstract concepts such as safety levels, collision risk, and the spatial clearance within which the UAV operates. Moreover, analyzing the spatial density of UAV visits enables interpretation of expert feature expectations through the framework of discounted state visitation frequency (SVF). In constrained environments, SVF is especially important as it effectively communicates and delineates policy tendencies toward establishing safe corridors or identifying hazardous zones. The Monte Carlo mean is employed as an approximation method to capture the distinct requirements of both experts μ_E and agents μ_π performing the task.

$$\mu_E \approx \frac{1}{N}\sum_{i=1}^{N}\sum_{t=0}^{T_i}\gamma^t\phi\left(s_t^i\right) \tag{6}$$

and

$$\mu_\pi \approx \frac{1}{M}\sum_{i=1}^{M}\sum_{t=0}^{T_j}\gamma^t\phi\left(s_t^j\right) \tag{7}$$

where, E represents the expert, while p_i denotes the agent. The variable T indicates the length of the trajectory. The parameters N and M correspond to the numbers of samples being evaluated. The state at time t in the ith trajectory is represented by s. The function ϕ describes the state at time t, and γ acts as the dependency coefficient. According to the maximum entropy principle, among all trajectory distributions consistent with the expert data, the one with the highest entropy is preferred to minimize unwarranted bias. Consequently, the probability $p_\theta(\tau)$ assigned to a given trajectory is defined as following Eq. (8):

$$p_\theta(\tau) = \frac{1}{Z(\theta)}\exp\left(\sum_{t=0}^{T} r_\theta(s_t, a_t)\right) \tag{8}$$

where $p_\theta(\tau)$ is the probability of the trajectory corresponding to the expert orbit τ is being considered. $Z(\theta)$ is the normalization coefficient. θ is the parametric vector of the reward function. The log-likelihood function of the expert dataset is defined as:

$$L(\theta) = \tau \in \sum_{\tau \in D_E} log\ p_\theta(\tau) \tag{9}$$

The task is to force the reward network to make the expert trajectory have the greatest probability. The gradient function ∇_θ is expressed as follows:

$$\nabla_\theta L(\theta) = \mu_E - \mu_\pi \tag{10}$$

This indicates that learning the reward function is equivalent to the process of tuning the reward network so that the distribution visits the state of the expert's distribution asymptomatic agent.

3.3 Soft Actor–Critic Agent

The Deep-IRL framework employs the SAC algorithm as its primary method for policy optimization. After each update to the reward network, SAC is trained to generate navigation trajectories based on the updated reward function. This iterative training occurs at every update cycle. Specifically, in each outer loop, the reward network is first updated by using the difference between the expert's and the agent's characteristics, and then the SAC agent performs multiple policy optimization steps under the updated reward. The trajectories produced by these two components gradually converge toward expert-level behavior, providing the necessary state visitation distribution for subsequent reward network updates. This alternating optimization explicitly couples reward inference and policy learning, ensuring that both components are iteratively aligned toward expert behavior. Through this dual-loop architecture, the system jointly refines both the latent reward function and the control policy, ultimately producing UAV's navigation strategies that are safe, smooth, and highly generalizable within constrained environments. The deep reward network defines a latent reward function, framing the navigation task as a continuous reinforcement learning problem. In this setting, the agent must learn to operate effectively within confined, obstacle-dense flight zones while maintaining stability and ensuring generalizability. It is important to note that optimizing the total reward alone may lead to irregular trajectories and localized congestion. To mitigate this, the SAC algorithm utilizes the maximum entropy principle to balance reward maximization with sufficient policy stochasticity. According to SAC, the learning objective is formally defined as follows:

$$\pi^* = \underset{\pi}{argmax} E_{\tau \sim \pi}\left[\sum_{t=0}^{T} \gamma^t \left(r_\theta(s_t, a_t) + \alpha H(\pi(\cdot \mid s_t))\right)\right] \tag{11}$$

where the reward function $r_\theta(s_t, a_t)$ is derived from the Deep-IRL network, capturing the safety and suitability of actions within their respective states. The term $\alpha H(\pi(\cdot \mid s_t)$ represents the policy entropy, which quantifies the randomness inherent in action selection, while α denotes the temperature coefficient that adjusts the relative importance of the entropy term. Incorporating entropy encourages the agent to maintain behavioral diversity, a crucial factor in constrained environments to avoid becoming trapped in local areas or hazardous passages. The policy is parameterized by an actor network $\pi_\phi(a \mid s)$ that directly maps the UAV's state to a probability distribution over a continuous action space. In this framework, the actor outputs the parameters of a Gaussian distribution, from which actions are sampled and then transformed into the valid action domain using the hyperbolic tangent (Tanh) function. This approach generates continuous and smooth control commands while preserving differentiability throughout the

sampling process. Functionally, the actor acts as a high-level controller, determining the flight direction and control intensity appropriate to the local environmental context.

To evaluate the quality of the actions proposed by the actor, the SAC algorithm employs two independent critic networks of $Q_{\psi 1}(s,a)$, and $Q_{\psi 2}(s,a)$. Each critic estimates the expected cumulative future reward when the UAV executes action a in state s and subsequently follows the current policy. The utilization of two critics facilitates the derivation of more conservative value estimates during training, thereby mitigating the risk of overestimation. This is particularly critical in constrained environments, where overestimation can precipitate high-risk decisions.

The update objective for the critics is formulated based on the soft Bellman equation [36]:

$$y = r_\theta(s,a) + \gamma\left(\underset{i=1,2}{min} Q_{\overline{\psi}_i}(s',a') - \alpha log\,\pi_\phi(a' \mid s')\right) \tag{12}$$

where s' denotes the subsequent state, a' is an action sampled from the policy p_i at state s', and $Q_{\overline{\psi}_i}$ is the target critic network, which is updated softly to enhance training stability $\alpha log\,\pi_\phi(a' \mid s')$. By incorporating entropy directly into the value estimation, the critic evaluates not only the immediate reward but also the policy's future flexibility.

Accordingly, the critics are trained to minimize the mean squared error between their predicted Q-values and the Bellman target. Concurrently, the actor is optimized to maximize the value estimated by the critics while maintaining a sufficiently high entropy level. This approach encourages the actor to increase the likelihood of actions favored by the critics without rendering the policy overly deterministic. Such a mechanism enables UAVs to prioritize safe and efficient navigation while retaining adaptability to novel spatial environments. During training, the heat coefficient is not fixed; instead, it is automatically adjusted to maintain the policy entropy at a predefined target level. This adaptive tuning obviates the need for manual parameter adjustment across different environments, thereby facilitating a more effective balance between exploration and exploitation.

3.4 Training Algorithms

In this section, we describe the algorithm used to train the IRL framework for UAV navigation in a confined space environment. The algorithm features a two-loop nested structure: the outer loop infers the latent reward function from expert data, while the inner loop optimizes the UAV control policy using SAC based on the current reward definition. Specifically, the outer loop updates the reward function based on expert demonstrations, while the inner loop performs multiple SAC optimization steps under the current reward.

The discrepancy between μ_E and μ_π reflects the difference in behavior exhibited by the UAV and the expert within the state space. The parameters of the reward network are updated following the IRL's principles:

$$\theta \leftarrow \theta + \alpha\left(\mu_E - \mu_\pi\right) \tag{13}$$

where α represents the learning rate. Intuitively, this update means that if the UAV rarely visits regions frequented by the expert, the rewards assigned to those regions increase. Conversely, if the UAV often enters areas the expert avoids, the corresponding rewards decrease. Over time, this iterative adjustment of the reward function aligns with the primary goal of ensuring safe navigation within constrained spatial environments.

Under the revised reward function, the SAC agent is trained at each iteration, incorporating each updated reward network. This iterative process, known as co-coaching, allows the reward function to progressively align more closely with expert behavior while the policy simultaneously evolves to optimize performance under this reward structure. This two-step procedure effectively separates the challenges of reward inference and policy optimization—despite their conceptual similarities, thereby reducing the complexity of addressing both simultaneously. The algorithm iterates until at least one convergence criterion is met, typically when the deviation in expected feature counts falls below a predefined threshold ε:

$$\|\mu_E - \mu_\pi\|_2 \leq \varepsilon \tag{14}$$

At convergence, the reward function and policy exhibit substantial consistency across successive iterations. The reward network is then considered a potential reward function that encapsulates the expert's safe navigation behavior, and the SAC policy is adopted as the UAV's control policy. By leveraging the deep-IRL algorithm, the system can retrospectively analyze and solve the problem of navigation within restricted areas, where manually specifying safety, smoothness, or obstacle avoidance criteria is challenging. Through iterative refinement of reward functions and policy optimization, the proposed methodology produces stable, secure, and highly generalizable flight strategies in complex environmental conditions. The training process and the setup of related parameters are carried out according to the pseudocode provided in Algorithm 1.

Algorithm 1: IRL-TP training process

Input:
Expert demonstrations $\mathcal{D}$
Learning rates α (reward), β (policy/critic)
Discount factor γ, entropy coefficient α_{ent}
Soft update coefficient τ, threshold ∈
Output:
Optimal policy πϕ*
Learned reward function r_θ

1: Initialize **reward network parameters θ**
2: Initialize **policy π_φ** and **critic networks Q_Ψ1, Q_Ψ2**
3: Initialize **target networks $\bar{Q}$_Ψ1, $\bar{Q}$_Ψ2 ← Q_Ψ1, Q_Ψ2**
4: Compute **expert feature expectation μ_E** from $\mathcal{D}$_E
5: **repeat** ▷ Outer loop (IRL)
6: Initialize **trajectory buffer $\mathcal{D}$_π ← ∅**
7: **for** k = 1 **to** K **do** ▷ Inner loop (SAC)
8: Observe current state **s_t**
9: Sample action **a_t ~ π_φ(a|s_t)**
10: Execute **a_t**, observe **s_{t + 1}**
11: Store transition **(s_t, a_t, s_{t + 1})** in replay buffer
12: Compute reward: **r_t ← r_θ(s_t, a_t)**

(Continued)

Algorithm 1 (continued)

Sample mini-batch **(s, a, s′)** from replay buffer
Sample next action **a′ ~ π_φ(a′|s′)**
Compute target:
**y ← r_t + γ (min(Q̄_Ψ1(s′, a′), Q̄_Ψ2(s′, a′))
−α_ent log π_φ(a′|s′))**
Update critics **Q_Ψ1, Q_Ψ2** by minimizing:
L_Q = (Q_Ψi(s, a) − y)^2
Update policy **π_φ** using:
∇_φ J = α_ent log π_φ(a|s) − Q_Ψ(s, a)
Soft update target networks:
Ψ̄_i ← τ Ψ_i + (1 − τ) Ψ̄_i
Append trajectory **τ** to **$\mathcal{D}$_π**
end for
Compute agent feature expectation:
μ_π ← E_{τ~$\mathcal{D}$_π} [Σ γ^t φ(s_t)]
Update reward network:
θ ← θ + α (μ_E − μ_π)
until **||μ_E − μ_π||$_2$ ≤ ε**
return **π_φ^*** and **r_θ**

4 Results and Discussion

Firstly, the authors collected and constructed an expert dataset within the simulator. The environmental scenarios were varied, each paired with corresponding state-action data. The dataset's parameters and details can be found at: https://github.com/buinghia3101/UAV_Dataset.git. All samples in the dataset are collected across 500 pre-calculated navigation scenarios. Along with that, the navigation range is divided into unit space zones with a size of 50 × 50 × 50. The process of data augmentation is done through the process of reversing the scenario direction and randomly rotating the navigation space. The model training and validation experiments were performed on a computer with an Intel Core i9 13900K CPU configuration, 64 Gb RAM, NVDIA GPU GTX3080. The platform used is Ubuntu 20.04 with Pybullet configuration, and Webots R2025a.

Fig. 3 illustrates the UAV's navigation trajectories operating within confined spatial environments, guided by the proposed Deep-IRL approach. In each subfigure, red hexagons represent obstacle regions or areas with high traversal costs, while the light blue grid indicates the permissible flight zone. Together, these color-coded elements form a planar pattern. Randomly generated noise zones are added to the environment to ensure action impact factors and scenario diversity. The mechanism of adding random obstacles is illustrated by the Eqs. (15) and (16):

$$O = \bigcup_{i=1}^{N_{obs}} \left\{ (c_i, r_i, z) \,|c_i \sim U\left(0, N_c - 1\right), \, r_i \sim U\left(0, N_r - 1\right), h_i \sim U\left(0, N_h - 1\right), z \in \left[h_i, h_i^{top}\right] \right\} \tag{15}$$

with

$$N_{obs} = \left[\rho \cdot N_c N_r N_h\right], \quad h_i^{top} \sim U\left(h_i, N_h - 1\right) \tag{16}$$

here, O is the set of obstacle tiles added to the scenario. N_c, N_r, N_h are the dimensions of the respective spatial reference dimensions (cols, rows, levers). ρ is the density factor. N_{obs} is the number of obstacles that

are randomly generated. The coordinates of the obstruction are denoted as c. The maximum elevation values of the obstacle cluster represented by h^{top}. The final smoothed UAV's flight path is shown as a dark blue line connecting the starting point to the target destination. Notably, the UAV successfully navigates through environments with high obstacle density without collisions. These trajectories ensure geometric feasibility and clearly demonstrate a strong tendency to avoid hazardous zones, maintain obstacle clearance, and sustain smooth, continuous motion. The learned reward function effectively captures the expert's behavioral priorities, including safety, avoidance of abrupt directional changes, and preference for clear flight corridors. The trajectory shape dynamically adapts to varying environmental configurations, reflecting the UAV's ability to adjust its path in response to structural constraints. Instead of following the shortest geometric path, the UAV generates soft, curved routes that pass through narrow passages within complex obstacle layouts. This behavior results from the entropy term in the SAC algorithm combined with the reward inference mechanism of the IRL framework, which together prevent the policy from converging to suboptimal local minima and encourage the selection of safer routes. Moreover, the resulting trajectories exhibit exceptional smoothness and stability, free from sudden oscillations or sharp turns, indicating that the established control policy aligns well with the UAV's kinematic requirements in constrained environments. Practical flight scenarios—such as corridors, industrial facilities, or densely populated urban areas—demand smooth and safe control maneuvers for effective deployment. The visual results confirm that the proposed method not only successfully infers the latent reward function from expert demonstrations but also leverages this knowledge to train a navigation policy capable of generalizing across diverse and complex environments. Altogether, these findings validate the effectiveness of the Deep-IRL framework combined with the SAC algorithm for UAV navigation in restricted spatial domains.

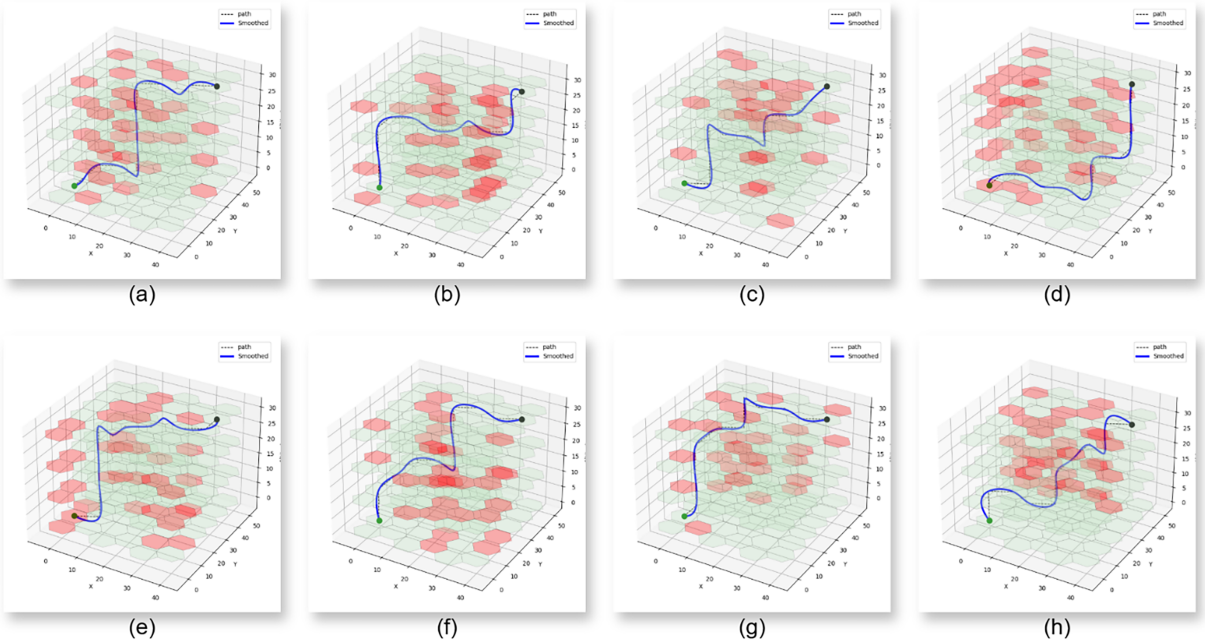

Figure 3: The UAV trajectory based on proposed IRL-TP framework, with the sample space planned according to hexagonal cells and the trajectory generated. The green cells represent free areas, while the red cells indicate areas where collisions are likely to occur based on the scenarios from (**a**) to (**h**) having the number of obstacles is randomly generated in a complex environment.

In Table 1, the proposed IRL-TP approach is quantitatively compared with typical approaches addressing the narrow UAV navigation problem in confined spaces. The proposed IRL-TP approach is quantitatively compared with typical approaches addressing the narrow UAV navigation problem in confined spaces. For clarity, the compared methods are grouped into three categories: classical planning methods (e.g., A, RRT-based), reinforcement learning methods, and imitation learning approaches. Evaluation metrics include trajectory length, energy consumption, execution time, orbital stability, and mission completion rate. Trajectory instability is defined as the cumulative deviation of the UAV's heading angle (or curvature) along the trajectory, reflecting motion smoothness. At first glance, the suggested approach achieved an average orbit length of 447.8 m and consumed no more than 1269 J of energy. In contrast, all traditional heuristic and RL-based methods fell significantly short of this performance. This indicates that the UAV's improved flight strategies were influenced by the reward function learned through IRL, which effectively minimized unnecessary movements and optimized spatial arrangements. Although the classical MIP (offline) method produced superior results in terms of orbit length and energy consumption, the proposed approach still performed very well, offering distinct advantages in online deployment and environmental friendliness. Demonstrating its ability to generate more direct and less convoluted routes in tight spaces, the proposed method achieved an average completion time of 82.1 s faster than most reinforcement learning and imitation learning methods. This is particularly important for real-world UAV missions, where flight time is closely linked to battery limitations and system safety. The stability index of the proposed method's trajectory was 0.044, significantly lower than that of traditional heuristic and RL methods, and closely approaching the classical optimal solution. This reflects the learned policy's geometric optimization and smoother, less oscillatory movements that better align with the UAV's kinematic constraints. A key advantage of the SAC combined IRL framework is its entropy component, which helps maintain smooth transition control without interruption. Despite a low failure rate, the proposed method achieved a success rate of 97.6%, outperforming traditional RL, imitation learning, and classical IRL methods. Overall, the proposed IRL-TP framework effectively combines classical optimization-based planners with learning-based techniques, achieving near-optimal performance comparable to established methods such as A* [20], RRT* [38], MIP [39], and DP [40]. At the same time, it maintains flexibility and real-time responsiveness typical of reinforcement learning and imitation learning frameworks. Unlike traditional methods that rely on offline computations and rigid environmental assumptions, this approach is applicable to a wide range of online scenarios and demonstrates robustness in dynamic and computationally constrained settings. Empirical results show that it consistently outperforms conventional RL and imitation learning (IL) methods in terms of safety margins, trajectory stability, and mission success rates. This improvement is attributed to reward modeling based on IRL, which provides more structured and informative learning signals. Additionally, IRL has been shown to enhance UAV navigation in constrained environments by significantly reducing uncertainty during exploration. Together, these findings highlight the effectiveness of the proposed methodology.

Table 1: Comparison between the proposed IRL-TP method with the state-of-the art methods.

Method	Trajectory Length (cm)	Energy (J)	Time (s)	Unstability	Success Rate (%)
Greedy planning [37]	525.6	1518	92.4	0.087	80.7
A* [18]	492.3	1421	86.9	0.074	85.2
RRT* [38]	501.8	1455	89.1	0.079	83.6
MIP (offline) [39]	438.7	1235	81.3	0.041	100
DP [40]	455.4	1296	83.7	0.048	96.1

(Continued)

Table 1 (continued)

Method	Trajectory Length (cm)	Energy (J)	Time (s)	Unstability	Success Rate (%)
Q-learning [41]	478.6	1384	87.5	0.066	88.9
DQN [42]	466.1	1337	85.9	0.061	91.4
PPO [43]	459.3	1312	84.6	0.055	93.2
BC [44]	471.9	1358	86.3	0.063	90.1
GAIL [12]	452.7	1291	83.9	0.050	95.0
RRT-RL [19]	456.2	1305	84.2	0.052	94.2
Proposed IRL-TP	447.8	1269	82.1	0.044	97.6

All methods used in the comparative experiments are configured to ensure fairness and stability (see Fig. 4). Whenever possible, core parameters such as batch size (e.g., time delay), number of training steps, episode count, and environment interaction limits are kept consistent. Stable performance is achieved by tuning parameters for reinforcement learning and imitation learning methods of GLC [37], GAIL [12], AIRL [13], and MaxEnt IRL [9] guided by recommendations from the original papers and preliminary experiments. However, it is important to note that parameter selection significantly impacts the performance of these baseline methods. Traditional RL techniques require manual design of the reward function and careful adjustment of balance coefficients among collision avoidance, goal achievement, and smoothness; otherwise, the learned policy may converge to local optima or unsafe behaviors. Adversarial IRL methods like GAIL [12] and AIRL [13] show considerable training variability due to their reliance on the discriminator's learning rate, update frequency, and overall entropy coefficient. In contrast, the proposed approach demands fewer parameters. By employing a deep reward network within the Max-Entropy IRL framework and using an improved SAC agent as the policy optimizer, the system autonomously balances exploitation and exploration through its entropy coefficient, reducing the need for manual tuning. In experiments, the same standard SAC parameters are effective across various environmental configurations with minimal adjustment. Additionally, initializing the policy on an expert trajectory significantly shortens the initial random exploration phase, accelerating convergence and making training less sensitive to the learning rate and discount factor. This advantage is evident in the smoother and more stable convergence curves of the proposed method compared to other IRL baselines. Overall, with only rough parameter tuning, the proposed baseline method achieves satisfactory performance using a fixed set of parameters.

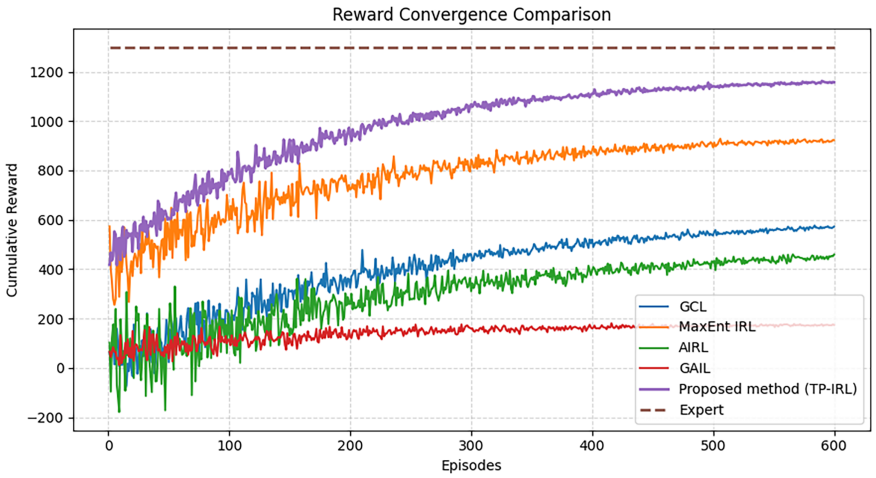

Figure 4: Comparison of the proposed method with RL-based methods in optimizing the reward function in the same amount of training parameters.

The authors conducted training and evaluation experiments on navigation methods using a deep learning framework under consistent scenario conditions. Table 2 presents quantitative results, offering a comprehensive assessment of UAV navigation methods based on various performance indicators, including work efficiency, safety level, movement quality, and training effectiveness. The comparison methods are grouped into two categories: reinforcement learning and imitation learning, and are analyzed accordingly. All methods were evaluated under identical environmental conditions and equivalent parameter tuning processes to ensure a fair comparison between different models. DQN achieved an 83.4% success rate, with a trajectory length of 132.6 m and a minimum obstacle distance of 0.41 m, serving as a baseline for comparison with other deep reinforcement learning methods. The use of PPO and SAC improved overall performance, achieving success rates of 84.9% and 87.6%, respectively, while reducing trajectory lengths to 125.3 m (PPO) and 121.8 m (SAC) and increasing minimum obstacle distances to 0.48 and 0.52 m. However, trajectory smoothness remained relatively high (SAC at 0.28), and the number of collisions was still notable (7 to 9 collisions per 100 episodes), indicating that while deep RL policies are effective, they are not always reliable in generating safe and wide trajectories in dense 3D environments. Deep imitation learning exhibited different behaviors. Behavior cloning achieved the fastest convergence (120 episodes) and lowest inference time (1.1 ms per step) but resulted in the longest trajectories (138.9 m), the lowest safety margin (0.39 m), and the highest collision rate (18% of episodes), leading to the lowest success rate (73.2%). This highlights the impact of distribution shift and error accumulation in purely supervised deep models. Although GAIL significantly improved success rates (85.1%) and reduced collisions to 8 per 100 episodes, it converged more slowly (780 episodes) and showed lower safety and smoothness metrics compared to the proposed method, indicating instability and high training complexity within deep imitation learning. In contrast, the proposed technique consistently outperformed all baselines across key metrics. It achieved the shortest trajectory length (112.2 m) and highest success rate, resulting in superior track efficiency. The minimum obstacle distance was significantly greater than SAC's, at 0.71 m, and trajectory smoothness was the lowest at 0.019. Additionally, it recorded only 2 collisions per 100 episodes, demonstrating high safety awareness and stable movement. The proposed method also converged rapidly, within approximately 340 training steps, much faster than GAIL and all deep reinforcement learning baselines, suggesting that incorporating expert trajectory structure reduces exploration burden and stabilizes model optimization. Although its inference time (2.6 ms per step) was slightly longer than PPO's, it remained within real-time operational limits. Overall,

experimental evidence shows that the proposed method consistently surpasses deep reinforcement learning and deep imitation learning baselines across all key metrics, delivering higher trajectory efficiency, safety margins, motion smoothness, and training stability even in dense 3D environments. By integrating expert trajectory structure into the learning process, this approach effectively reduces exploration demands and enhances policy robustness while maintaining reliable real-time UAV navigation.

Table 2: Comparison between the proposed IRL-TP method with the state-of-the-art methods using DL-based networks.

Method	Success Rate (%)	Trajectory Length (m)	Minimum Distance to Obstructions (m)	Orbital Smoothness	Number of Collisions/100 ep Episodes	Convergence Time (episodes)	Inference Time (ms/step) ↓
DQN [42]	78.4	132.6	0.41	0.037	14	~950	3.2
PPO [43]	84.9	125.3	0.48	0.031	9	~620	2.9
SAC [33]	87.6	121.8	0.52	0.028	7	~540	3.5
BC [44]	73.2	138.9	0.39	0.042	18	~120	1.1
GAIL [45]	85.1	124.6	0.50	0.030	8	~780	3.8
Proposed method	93.7	112.2	0.71	0.019	2	~340	2.6

Regarding the average corridor generation time, all methods exhibit an increasing trend as environmental complexity rises from simple to complex (see Fig. 5). Specifically, in the simple environment, the DQN [42] and PPO [43] methods require only about 0.15–0.18 s, whereas SAC [33] and GAIL [45] take significantly longer—approximately 0.82 and 0.76 s, respectively. The proposed method, TP-IRL, achieves a corridor generation time of around 0.32 s, which is substantially lower than SAC and GAIL, while maintaining stability with a small standard deviation. When transitioning to the medium environment, SAC and GAIL times increase by about 1.05 and 0.98 s, respectively, whereas TP-IRL only slightly rises to 0.41 s, indicating good scalability as spatial complexity grows. In the complex environment, SAC and GAIL further increase to roughly 1.18 and 1.12 s, respectively, while TP-IRL remains at 0.53 s, almost twice as long as standard deep RL techniques but still significantly more efficient. This demonstrates that the reward structure learned through IRL substantially reduces inference costs when generating corridors in narrow spaces. Considering the average safe area of the corridor, the proposed method consistently achieves the highest values across all three scenarios. In the simple environment, TP-IRL outperforms DQN (~42 m^2), PPO (~49 m^2), SAC (~62 m^2), and GAIL (~55 m^2), creating corridors with an average area of approximately 69–70 m^2. In the medium environment, the corridor areas of RL methods drop sharply, with DQN at about 18 m^2 and PPO at around 22 m^2. However, TP-IRL maintains a larger area of about 34 m^2, surpassing SAC (~29 m^2) and GAIL (~25 m^2). This trend continues in the complex environment, where DQN, PPO, and GAIL achieve roughly 9, 11, and 13 m^2, respectively, while TP-IRL attains about 20 m^2. These results indicate that the corridors generated by the proposed method not only persist in complex environments but also maintain a wider safety margin. Combining these findings, the proposed method clearly balances computational efficiency and geometric safety. While traditional RL and IL methods often trade off corridor generation time for safety or *vice versa*, TP-IRL simultaneously reduces corridor generation time by 40%–55% compared to SAC and GAIL and increases the safe area by approximately 15%–30% in complex environments. This confirms the advantage of integrating IRL into the RL framework, enabling the control policy to learn a more reasonable and stable trajectory structure for UAV navigation in narrow 3D environments.

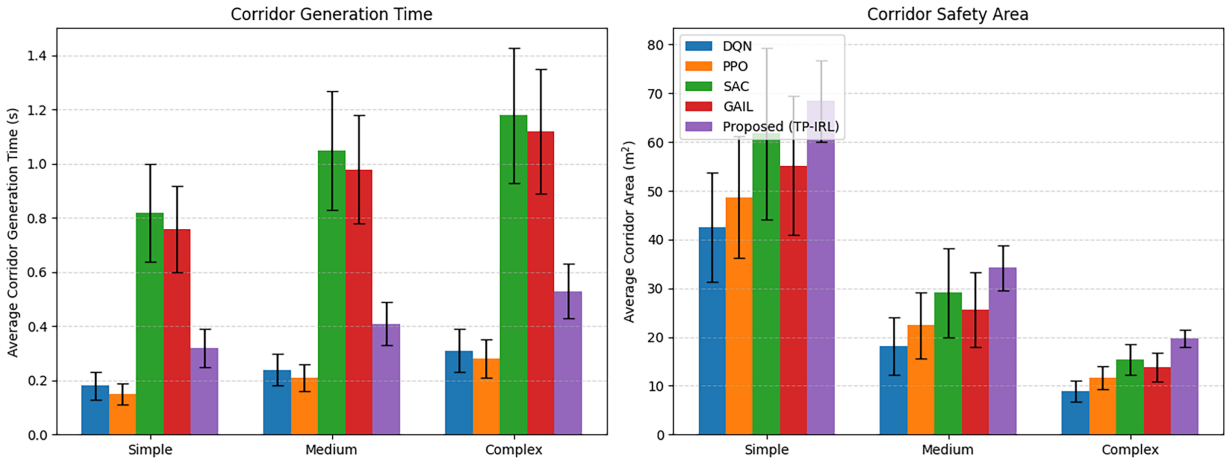

Figure 5: Comparison of average corridor generation time and average corridor safety area of DQN [42], PPO [43], SAC [33], GAIL [12,45], and proposed IRL-TP in three environmental levels of simple, medium, and complex.

Fig. 6 illustrates the results of UAV navigation in a realistic environment with few obstacles, while scenarios with dense obstacles are presented at two levels of difficulty. As shown in Fig. 6, the narrow obstructions created many open spaces, allowing for multiple possible routes. Under these conditions, it became clear that the UAV navigated toward the target with minimal abrupt changes in direction and trajectory. The UAV avoids obstacles at close range while maintaining directional accuracy during sharp turns or sudden stops, suggesting that the learned policy not only enhances target approach but also implicitly incorporates criteria such as kinematic stability and movement efficiency. Based on the reward function derived from the expert trajectory, this indicates that the UAV's focus on precise paths leads to behaviors prioritizing crash avoidance over control costs, thereby improving safety in less hazardous environments. In contrast, Fig. 7 shows an exponential increase in the number of obstacles, resulting in numerous narrow corridors, blocked areas, and zones with a high risk of collision. To navigate through these obstacles, UAVs must perform advanced maneuvers, such as continuously changing direction, reducing their turning radius, and adjusting their speed to match that of approaching objects. The outlined path demonstrates that the UAV deliberately selects feasible routes, shields itself from obstacles, and avoids areas with limited geometry—even when shorter paths are available. The learned reward function effectively encodes the impact of obstacle density on navigation, compelling UAVs to balance safety requirements with destination objectives in confined spaces. By comparing scenarios with sparse and dense obstacles, it is evident that obstacle density plays a crucial role in navigation behavior. Despite significant differences in environmental structure, the proposed method ensures consistent UAV guidance toward the target without requiring adjustments to the reward function or parameters. This demonstrates the generalizability of the inferred reward function, which reliably accounts for changes in obstacle density and adapts navigation strategies in narrow and complex environments using the Deep-IRL framework combined with SAC.

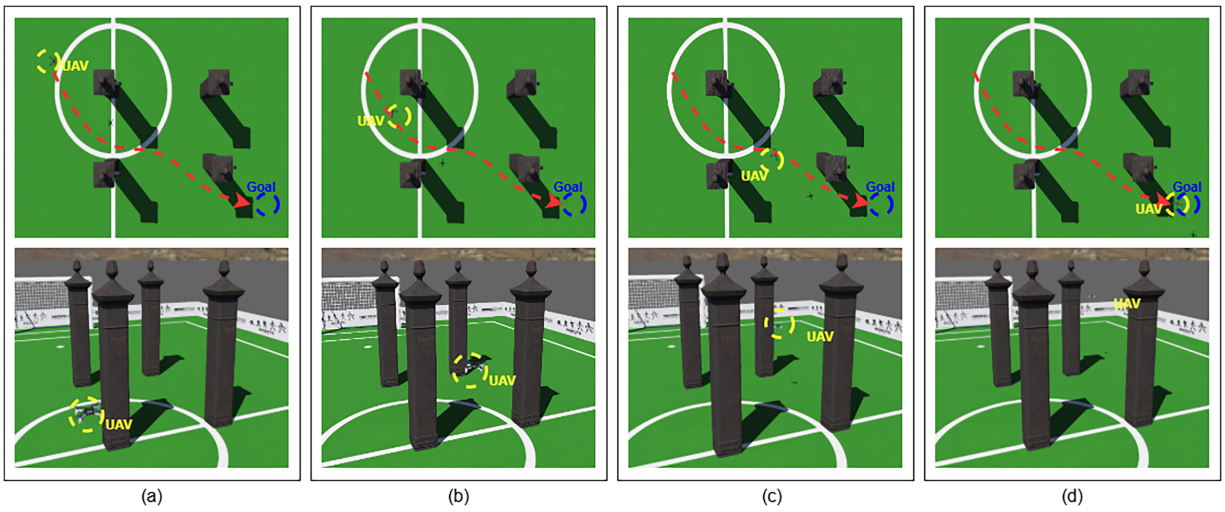

Figure 6: The experiment simulates navigation in an environment with moderately dense obstacles. The image pairs from top to bottom show the viewing angles, and from left to right, they are recorded over time in the sequence (**a**–**d**) respectively. The process illustration is illustrated as shown in the link: https://youtu.be/1EP-hFa1wrk.

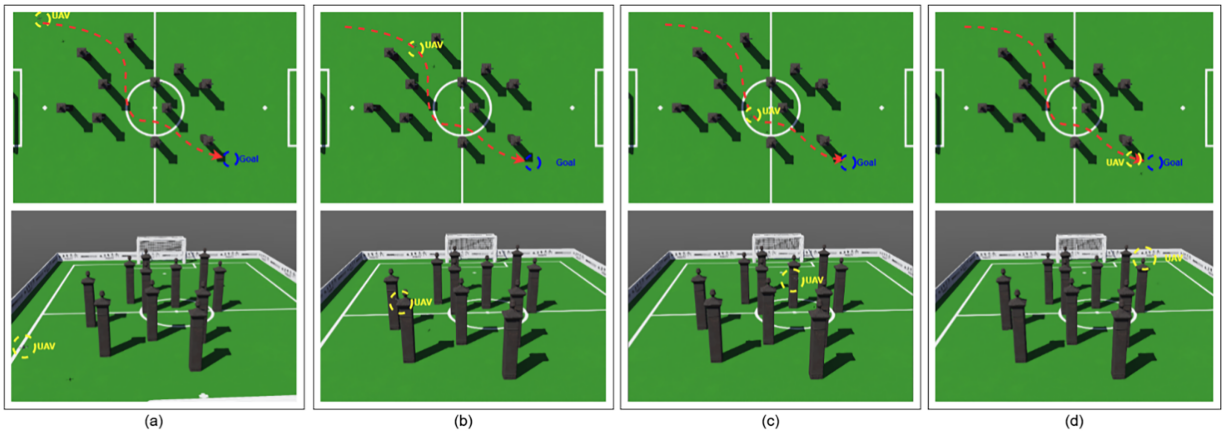

Figure 7: The experiment simulates navigation in a high-density obstacle environment. The image pairs from top to bottom show the viewing angles, and from left to right they are recorded over time in the sequence (**a**–**d**) respectively. The process illustration is illustrated as shown in the link: https://youtu.be/1EP-hFa1wrk.

The authors conducted experiments on a four-motor UAV frame platform (see Fig. 8). A visual SLAM framework is set up as the vision containing point clouds. The corridor area is established and planned for continuous navigation activities. The set of trajectories was collected and evaluated as shown in Figs. 8 and 9. The operation of the UAV in practice shows that there were no collisions with identified obstacles. Additionally, the rapid inference speed meets the real-time operation of the UAV on the Jetson Nano Origin embedded device. Through promising experimental results, the proposed method demonstrates its potential in solving UAV navigation tasks in obstacle-rich environmental scenarios as in the experimental conditions. The phenomena of shaking, deviation, and instability occur with limited frequency and mainly stem from the constrained hardware configuration. The trajectory planning method has been proposed with the aim of high applicability in emerging UAV applications. This method facilitates efficient and safe navigation in crowded airspace, especially under low-altitude flight conditions. By reducing mission costs, it also enhances

coverage efficiency and sensor quality for UAV data collection. Adaptive positioning for communication, edge computing, and task allocation is one of the capabilities it offers in new UAV computing models, making it a crucial element in future UAV technology.

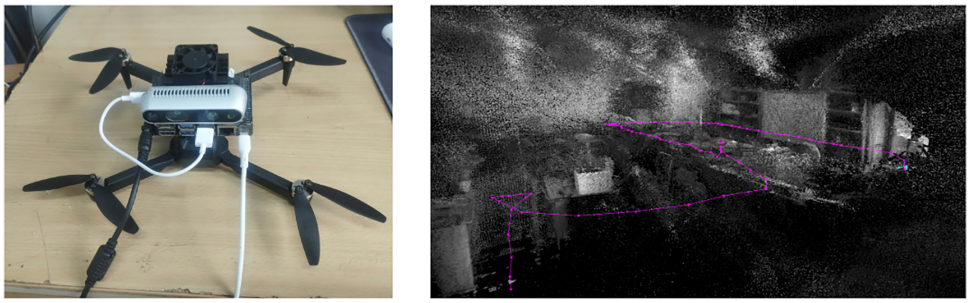

Figure 8: The UAV device frame is used for experiments, and the trajectory results are verified in the case. The UAV equipment used (**a**) and the estimated trajectory in the environmental scenario (**b**) are shown by the pink line.

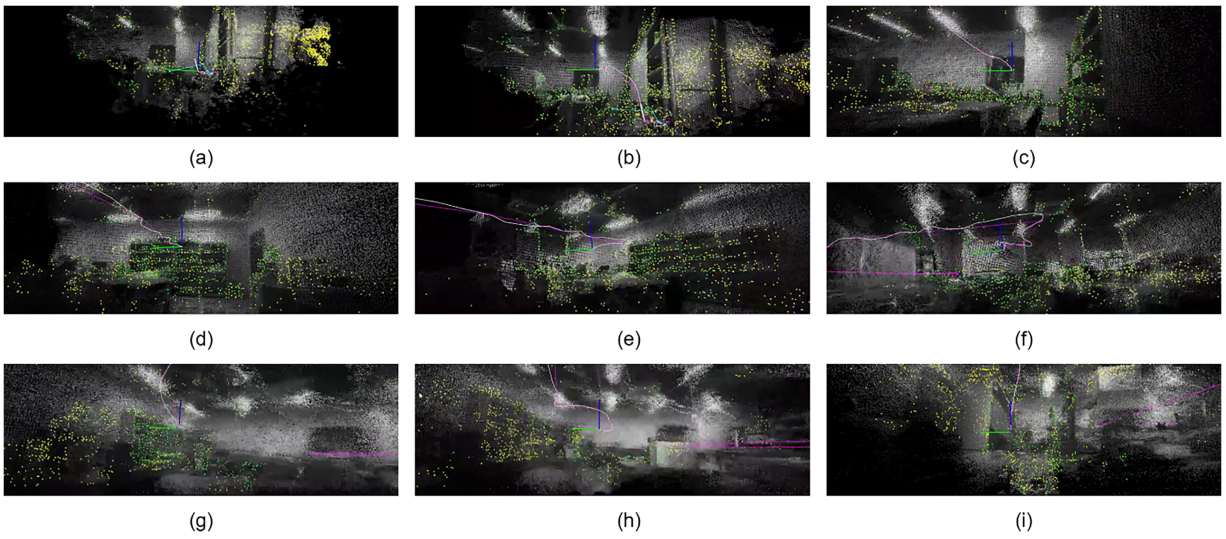

Figure 9: Results of tracking the UAV navigation process in the observed environment. In chronological order. The process of takeoff and entering a stable orbit (**a**–**c**). The process of changing the approach angle when nearing an obstacle (**d**–**f**). The process of stabilizing position and landing (**g**–**i**). The constructive trajectories were marked in pink throughout the experiment.

5 Conclusions

The paper proposes the novel deep-IRL-TP framework to solve the problem of navigation by UAVs in narrow space conditions with high obstacle density. In contrast to traditional reinforcement learning methods that rely on manual calculation of reward functions, the proposed approach directly derives an implicit compensation function by exploiting expert trajectories through a deep reward network. This implies that the implicit reward function is indicative of navigation goals such as safety, trajectory smoothness, and movement efficiency. Using the learned reward function as a foundation, the SAC agent is integrated to optimize the UAV control policy in continuous space, while maintaining an equilibrium between exploitation and exploration through the regularization of entropy. Furthermore, the hexagonal grid-based

spatial representation introduced provides uniform angular resolution and reduces directional bias, thereby improving the consistency and efficiency of navigation planning in complex environments. In addition, the adoption of the Maximum Entropy principle enables a more optimal policy learning process by balancing exploration and exploitation, resulting in improved training stability and more robust trajectory generation. By examining simulation outcomes in various scenarios with different obstacle densities, the proposed framework is demonstrated to have not only a high task completion rate and shorter trajectories but also demonstrates superior stability and generalizability compared to existing methods of RL, imitation learning, and IRL. Environmental tests show that the proposed method is highly competitive with approaches in the same scenario. The success rate reached a high of 97.6%, while the number of episodes needed to converge the parameters was much faster than other methods (~340). Accuracy and safety parameters in the navigation process are significantly superior to those of conventional RL and IL models. While the inference process ensures real-time accessibility (2.6 ms). Even when confronted with a dense and convoluted environment, our technique maintains streamlined navigation behavior, effectively prevents collisions, and converges on a dependable policy, demonstrating the superiority of learning an implicit reward function over manual design. Future research will include extending the proposed framework to multi-UAV contexts, dynamic environments with moving obstacles, and access to real-world sensor data (RGB–D, LiDAR) so that it can be deployed on physical UAV platforms. Additionally, sim-to-real learning and multi-agent IRL will likely improve the system's adaptability and reliability in real-world scenarios.

Acknowledgement: Not applicable.

Funding Statement: Not applicable.

Author Contributions: Conceptualization, Xuan-Thuc Nguyen; Methodology, Xuan-Thuc Nguyen, Le-Minh Nguyen; Software, Nhu-Nghia Bui, Thai-Viet Dang; Formal analysis, Ngoc-Quynh Nguyen; Investigation, Xuan-Thuc Nguyen, Thai-Viet Dang; Resources, Nhu-Nghia Bui, Thai-Viet Dang; Data curation, Dinh-Quy Vu, Thai-Viet Dang; Writing—original draft, Xuan-Thuc Nguyen, Dinh-Quy Vu, Thai-Viet Dang; Writing—review & editing, Dinh-Quy Vu, Thai-Viet Dang; Visualization, Dinh-Quy Vu, Thai-Viet Dang; Supervision, Xuan-Thuc Nguyen, Thai-Viet Dang; Project administration, Xuan-Thuc Nguyen. All authors reviewed and approved the final version of the manuscript.

Availability of Data and Materials: The data that support the findings of this study are available on request from the corresponding author, Thai-Viet Dang.

Ethics Approval: Not applicable.

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

References

1. Sheltami T, Ahmed G, Ghaleb M, Mahmoud A. UAV path planning and trajectory optimization: a comprehensive survey. Arab J Sci Eng. 2026;51(1):105–45. doi:10.1007/s13369-025-10971-8.
2. Han H, Cheng J, Lv M, Xi Z. Autonomous navigation of UAVs in unknown 3D environments using deep reinforcement learning for path planning. IEEE Trans Veh Technol. 2025;74(11):16894–907. doi:10.1109/TVT.2025.3581333.
3. Xue Y, Chen W. Combining motion planner and deep reinforcement learning for UAV navigation in unknown environment. IEEE Robot Autom Lett. 2024;9(1):635–42. doi:10.1109/LRA.2023.3334978.
4. Wu JJ, Tseng KS. Adaptive submodular inverse reinforcement learning for spatial search and map exploration. Auton Rob. 2022;46(2):321–47. doi:10.1007/s10514-021-10025-6.
5. Chan JH, Liu K, Chen Y, Sagar ASMS, Kim YG. Reinforcement learning-based drone simulators: survey, practice, and challenge. Artif Intell Rev. 2024;57(10):281. doi:10.1007/s10462-024-10933-w.

6. Peng S, Luo Z, Yang L, Jiang W, Yu S. Aerial-ground collaborative mapping and path planning algorithm for unmanned systems in dynamic environments. Int J Intell Robot Appl. 2026;10(1):345–59. doi:10.1007/s41315-025-00503-w.
7. Norouzi P, Shahbazi H, Torabi K. Optimizing quadrotor navigation through emotional deep reinforcement learning: leveraging emotional rewards and states for enhanced training efficiency. J Braz Soc Mech Sci Eng. 2026;48(2):115. doi:10.1007/s40430-025-06091-x.
8. Ruiz-Serra J, Harré MS. Inverse reinforcement learning as the algorithmic basis for theory of mind: current methods and open problems. Algorithms. 2023;16(2):68. doi:10.3390/a16020068.
9. Song L, Guo Q, Ali Channa I, Wang Z. A survey of maximum entropy-based inverse reinforcement learning: methods and applications. Symmetry. 2025;17(10):1632. doi:10.3390/sym17101632.
10. You C, Lu J, Filev D, Tsiotras P. Advanced planning for autonomous vehicles using reinforcement learning and deep inverse reinforcement learning. Robot Auton Syst. 2019;114(5):1–18. doi:10.1016/j.robot.2019.01.003.
11. Wu Z, Sun L, Zhan W, Yang C, Tomizuka M. Efficient sampling-based maximum entropy inverse reinforcement learning with application to autonomous driving. IEEE Robot Autom Lett. 2020;5(4):5355–62. doi:10.1109/LRA.2020.3005126.
12. Huang H, Li J, Lu F, Li J. Generative adversarial imitation learning computing task offloading scheme for optimizing of generated sample utilization and system overhead. J Supercomput. 2024;81(1):289. doi:10.1007/s11227-024-06744-z.
13. Wang H, Liu X, Zhou X. Autonomous UAV interception via augmented adversarial inverse reinforcement learning. In: Proceedings of 2021 International Conference on Autonomous Unmanned Systems (ICAUS 2021). Singapore: Springer Singapore; 2022. p. 2073–84.
14. Houghton MD, Oshin AB, Acheson MJ, Theodorou EA, Gregory IM. Path planning: differential dynamic programming and model predictive path integral control on VTOL aircraft. In: Proceedings of the AIAA SCITECH 2022 Forum; 2022 Jan 3–7; San Diego, CA, USA.
15. Cao K, Xu X, Jin W, Johansson KH, Xie L. A differential dynamic programming framework for inverse reinforcement learning. IEEE Trans Robot. 2025;41:6267–86. doi:10.1109/TRO.2025.3623769.
16. Song W, Tong S. Inverse reinforcement learning optimal control for Takagi-Sugeno fuzzy systems. Artif Intell Sci Eng. 2025;1(2):134–46. doi:10.23919/AISE.2025.000010.
17. Song L, Ali Channa I, Wang Z, Sun G. Dynamic heterogeneous multi-agent inverse reinforcement learning based on graph attention mean field. Symmetry. 2025;17(11):1951. doi:10.3390/sym17111951.
18. Dang TV, Tan PX. Hybrid mobile robot path planning using safe JBS-A*B algorithm and improved DWA based on monocular camera. J Intell Rob Syst. 2024;110(4):151. doi:10.1007/s10846-024-02179-z.
19. Pham HL, Bui NN, Dang TV. Hybrid path planning for wheeled mobile robot based on RRT-star algorithm and reinforcement learning method. J Robot Control. 2025;6(4):2045–51. doi:10.18196/jrc.v6i4.27678.
20. Bui TL, Nguyen DQ, Luu VH, Phan DH, Dang TV. TC-SRRT*HAC: a HAC controller-based trackability-constrained spatio-temporal RRT* for manipulator motion planning in dynamic environments. J Appl Sci Eng. 2026;31:26031007. doi:10.6180/jase.202608_31.007.
21. Hu L, Kong Z. Multi-target point path planning algorithm for mobile robot based on probabilistic roadmap. Intell Serv Robot. 2025;19(1):9. doi:10.1007/s11370-025-00670-6.
22. Tao F, Chen Z, Wang Z, Zhu L, Wang J. Multistrategy improved particle swarm optimization algorithm for path planning of UAV in 3-D low altitude urban environment. IEEE Internet Things J. 2025;12(19):40470–83. doi:10.1109/JIOT.2025.3589644.
23. Gai W, Zheng Y, Zhang J, Zhang G. A novel leader-follower-based hybrid particle swarm-grey wolf optimizer algorithm for the constrained UAV path planning. Aircr Eng Aerosp Technol. 2025;97(5):636–47. doi:10.1108/aeat-08-2024-0232.
24. Yin S, Yang J, Ma L, Fu M, Xu K. An enhanced whale algorithm for three-dimensional path planning for meteorological detection of the unmanned aerial vehicle in complex environments. IEEE Access. 2024;12(1):60039–57. doi:10.1109/ACCESS.2024.3394055.

25. Zhang H, Guo C, Zhai D, Wang Y, Liu H, Chen F, et al. Application of improved crown porcupine optimizer in UAV path planning based on dynamic weighted JAYA-CPO attack strategy. Prot Control Mod Power Syst. 2025;10(6):101–27. doi:10.23919/PCMP.2024.000413.
26. Sharma VC, Roy S. An overview of Q-learning and deep Q-learning for an autonomous multi-UAV wireless network. Expert Syst. 2025;42(11):e70143. doi:10.1111/exsy.70143.
27. Adhikari B, Khwaja AS, Jaseemuddin M, Anpalagan A, Nallanathan A. Energy efficient RIS-assisted UAV networks using twin delayed DDPG technique. IEEE Trans Wirel Commun. 2024;23(12):18423–39. doi:10.1109/TWC.2024.3468162.
28. Galvez-Serna J, Vanegas F, Brar S, Sandino J, Flannery D, Gonzalez F. UAV4PE: an open-source framework to plan UAV autonomous missions for planetary exploration. Drones. 2022;6(12):391. doi:10.3390/drones6120391.
29. Li T, Huai T, Li Z, Gao Y, Li H, Zheng X. SkyVLN: vision-and-language navigation and NMPC control for UAVs in urban environments. arXiv:2507.06564. 2025.
30. Chen K, Fang X, Ren C, Jiang H, Li B. Recursive neural network-based design of unmanned aircraft swarm collaborative mission execution and autonomous navigation system. Appl Math Nonlinear Sci. 2025;10(1):20250772. doi:10.2478/amns-2025-0772.
31. Wei A, Liang J, Lin K, Li Z, Zhao R. DTPPO: dual-transformer encoder-based proximal policy optimization for multi-UAV navigation in unseen complex environments. Drones. 2024;8(12):720. doi:10.3390/drones8120720.
32. Guo J, Zhou G, Huang H, Huang C. Advancements in UAV path planning: a deep reinforcement learning approach with soft actor-critic for enhanced navigation. Unmanned Syst. 2025;13(4):1065–84. doi:10.1142/s2301385025500669.
33. Wang Y, Zhao H, Huang H, Li D, Ni Y, Gui G. Multi-task multi-agent reinforcement learning for collaborative radio mapping and navigation in cellular-connected UAV networks. IEEE Trans Cogn Commun Netw. 2026;12:4731–45. doi:10.1109/TCCN.2025.3641516.
34. Zhang J, Yu Z, Mao S, Periaswamy SCG, Patton J, Xia X. IADRL: imitation augmented deep reinforcement learning enabled UGV-UAV coalition for tasking in complex environments. IEEE Access. 2020;8:102335–47. doi:10.1109/ACCESS.2020.2997304.
35. Kou K, Yang G, Zhang W, Yao Y, Zhou X. UAV autonomous navigation with hybrid maneuver modes: a hierarchical reinforcement learning method. Intell Data Anal Int J. 2026;3:1088467X251408949. doi:10.1177/1088467x251408949.
36. Tan J, Chen G, Huang Z, Liu H, Ang MH Jr. E-GAIL: efficient GAIL through including negative corruption and long-term rewards for robotic manipulations. Appl Intell. 2025;55(7):633. doi:10.1007/s10489-025-06335-2.
37. Huang T, Fan K, Sun W, Li W, Guo H. Potential-field-RRT: a path-planning algorithm for UAVs based on potential-field-oriented greedy strategy to extend random tree. Drones. 2023;7(5):331. doi:10.3390/drones7050331.
38. Wu C, Guo Z, Zhang J, Mao K, Luo D. Cooperative path planning for multiple UAVs based on APF B-RRT* algorithm. Drones. 2025;9(3):177. doi:10.3390/drones9030177.
39. Chen Z, Wang S, Chen K, Zhang X. Probability-constrained path planning for UAV logistics using mixed integer linear programming. Modelling. 2025;6(3):82. doi:10.3390/modelling6030082.
40. Lin CE, Syu YM. GA/DP hybrid solution for UAV multi-target path planning. J Aeronaut Astronaut Aviat. 2016;48(3):203–20. doi:10.6125/16-0704-894.
41. Li D, Yin W, Wong WE, Jian M, Chau M. Quality-oriented hybrid path planning based on A* and Q-learning for unmanned aerial vehicle. IEEE Access. 2022;10:7664–74. doi:10.1109/ACCESS.2021.3139534.
42. Samma H, El-Ferik S. Autonomous UAV visual navigation using an improved deep reinforcement learning. IEEE Access. 2024;12:79967–77. doi:10.1109/ACCESS.2024.3409780.
43. Su M, Chai H, Zhao C, Lyu Y, Hu J. Lightweight obstacle avoidance for fixed-wing UAVs using entropy-aware PPO. Drones. 2025;9(9):598. doi:10.3390/drones9090598.
44. Wei P, Liang R, Michelmore A, Kong Z. Vision-based 2D navigation of unmanned aerial vehicles in riverine environments with imitation learning. J Intell Rob Syst. 2022;104(3):47. doi:10.1007/s10846-022-01593-5.
45. Jiang S, Ge Y, Yang X, Yang W, Cui H. UAV control method combining reptile meta-reinforcement learning and generative adversarial imitation learning. Future Internet. 2024;16(3):105. doi:10.3390/fi16030105.