Journal of Systems Architecture 168 (2025) 103566

Contents lists available at ScienceDirect

Journal of Systems Architecture

journal homepage: www.elsevier.com/locate/sysarc

Trajectory design for data collection under insufficient UAV energy: A staged actor–critic reinforcement learning approach

Jing Mei [a], Yuejia Zhang [a], Zhao Tong [a,*,1], Keqin Li [b,c,2]

[a] College of Information Science and Engineering, Hunan Normal University, Changsha, 410081, Hunan, China
[b] College of Information Science and Engineering, Hunan University and National Supercomputing Center in Changsha, 410082, Hunan, China
[c] Department of Computer Science, State University of New York, 12561, NY, New Paltz, USA

ARTICLE INFO

Keywords:
Data collection
DRL
Energy efficiency
Fixed-wing UAV
Staged actor–critic based reinforcement learning (S-ACL)

ABSTRACT

Fixed-wing unmanned aerial vehicles (UAVs) offer distinct advantages for large-scale environmental sensor data collection. In forest and marine scenarios, UAVs typically depart from a fixed location, collecting data along a route, and return. Unlike existing work aiming to minimizing energy consumption on data collection task, this study focus on the scenario where a UAV's initial energy may not be sufficient to visit all sensor nodes. We aim to maximize data collection under insufficient battery energy while make a safety return. To solve this, we adopt the twin delayed deep deterministic policy gradient (TD3) algorithm with three designed reward functions, and introduce a stage-based safe action algorithm, termed Staged Safe-Action TD3 (SS-TD3). An energy consumption model incorporating acceleration and a segmented time model are used to enhance exploration efficiency. To tackle sparse binary rewards and the suboptimal convergence of complex reward function in reinforcement learning, a staged training approach, Staged Actor–Critic based reinforcement Learning (S-ACL) is proposed, as the one of the fundamental component of SS-TD3. Experimental results show that SS-TD3 achieves the best energy efficiency compared to baselines, while S-ACL significantly improves policy performance in complex reward environments.

1. Introduction

In recent years, unmanned aerial vehicles (UAVs) have attracted widespread attention as flying wireless communication platforms due to the high mobility and flexible deployment capabilities [1,2]. In cellular applications, UAVs have been utilized as temporary base stations to rapidly restore service during natural disasters [3] and to offload data during special events or high-traffic periods [4]. In the context of the Internet of Things (IoT) and wireless sensor networks (WSNs), UAVs also serve as mobile aggregators, enabling efficient data collection from ground-based sensor devices [5].

Since UAVs are fully controllable and capable of operating at the air, they can be dispatched to perform specialized missions in remote and challenging regions, such as forests, plateaus, polar areas, oceans, and deserts. These regions are often characterized by harsh environmental conditions, vast geographic coverage, and prohibitively high costs for establishing reliable ground-based communication infrastructure. Wireless sensors (SNs) are commonly deployed there for purposes such as meteorological monitoring, structural health monitoring of buildings, and environmental and ecological monitoring. These sensors typically operate on a periodic or event-triggered basis, collecting data at fixed intervals or in response to specific events. These sensors are often sparsely distributed in the monitored area and are characterized by relatively low generated data volume, high latency tolerance, and extreme energy efficiency [6].

Given the agility, on-demand deployment capabilities, low-altitude operation, and the ability to establish reliable communication links with ground sensors, UAVs are extensively used for data collection tasks, as the preferred solution in many remote monitoring and data gathering scenarios [7–9].

1.1. Related works

Despite of the advantages, the limited battery capacity of UAVs poses significant challenges to practical applications. In remote regions,

* Corresponding author.
E-mail addresses: jingmei@hunnu.edu.cn (J. Mei), 202320294086@hunnu.edu.cn (Y. Zhang), tongzhao@hunnu.edu.cn (Z. Tong), lik@newpaltz.edu (K. Li).
[1] Member, IEEE.
[2] Fellow, IEEE.

https://doi.org/10.1016/j.sysarc.2025.103566
Received 11 April 2025; Received in revised form 8 July 2025; Accepted 22 August 2025
Available online 15 September 2025

UAVs assigned to collect sensor data must return to areas with reliable network infrastructure. The long-distance round trips required for such missions consume a substantial amount of battery energy. Furthermore, UAVs must also be retrieved after completing the tasks. Consequently, efficient utilization of UAV battery energy necessitates careful trajectory optimization or deployment planning. Sun et al. [10] applied the deep deterministic policy gradient (DDPG) algorithm to optimize UAV trajectories, maximizing data collection while minimizing flight energy consumption. Fang et al. [11] reduced the total energy consumption of the offloading system by jointly optimizing the task offloading decision, 3D deployment of two-tier UAVs, the elevation angle of the bottom UAV. Kang et al. [12] employed UAVs for multi-hop routing in IoT systems to aggregate and collect data, achieving optimized total system energy consumption.

While the studies just mentioned and [13,14] directly or indirectly improved UAV flight energy efficiency, they did not account for scenarios involving insufficient energy or the optimization of energy required for the return trip. Xiuwen et al. [15] introduce the unmanned ground vehicle as a charging platform for the UAV assisted IoT networks, optimize the cooperative trajectories and enabling the UAV to take off and land on the UGV at appropriate times. Mekala et al. [16] maximize the data collection and reduce the AoI of collected data under the energy consumption constraint of UAVs in the context of urban flood monitoring. However, our research is a bit more challenge. The SNs are self-sustaining, and the UAV dones't have sufficient battery energy to perform the usual energy optimization, which maximize the total data collection objective at first, and optimize energy consumption next.

In remote regions characterized by large geographic coverage, deploying multiple UAVs performing data collection tasks is a common approach. Studies such as [10,12] explored the use of multiple UAVs for data collection. However, these methods require addressing challenges such as collision avoidance and other additional factors, resulting in complex trajectory planning and high deployment difficulties.

An alternative strategy is to establish multiple operational docks, with each dock responsible for a specific area. UAVs can then be deployed on demand to collect data within the assigned areas before returning to the respective dock for recharging or further deployment. Guo et al. [17] investigated UAV-assisted downlink wireless networks with fixed charging stations, optimizing charging service time allocation, flight trajectories, and transmission power allocation to maximize user data rates. Li et al. [18] optimized UAV flight distances and minimized task completion times. A V-shaped trajectory in large-scale areas is proposed by taking into account that UAVs can continue collecting data while flying away from SNs during hovering-based data collection.

During each flight, the maximum available energy of a fully charged UAV is generally a constant, while the minimum energy required to cover a target area remains uncertain. This study focuses on data collection tasks performed by a single UAV within a predefined target area. Specifically, we aim to design UAV trajectories that adapt to energy constraints, optimizing energy efficiency under conditions of insufficient energy availability.

Existing research predominantly considers two types of UAVs: rotary-wing UAVs and fixed-wing UAVs. Most of the aforementioned studies focus on the former. Compared to rotary-wing UAVs with hovering ability, fixed-wing UAVs offer greater payload capacity, higher speeds, and longer operational lifespans. They are more appropriate for long-distance missions or applications involving remote WSNs from this perspective, and is often given priority selection in such scenarios [19–21]. Also hybrid-wing UAV combine the advantages of both rotary-wing and fixed-wing, but is relatively expensive [22]. In this study, we focus on the use of fixed-wing UAVs for round-trip data collection in remote region, aiming to address the challenges of optimizing energy efficiency in such scenarios.

Due to the sparse distribution of SNs, applying DRL in such a sparse reward environment often encounter the suboptimal performance, which require additional approach to address [23–25]. Heuristic algorithms are generally less impacted by the sparsity of SNs and are therefore widely applied to UAV trajectory planning tasks in remote areas [18,19]. However, as trajectory guidance methods, their solution lie on discrete space. In contrast, intelligent agents trained using reinforcement learning (RL) can continuously and adaptively adjust UAV motion strategies, adapt to continuous space, enabling obstacle avoidance [26] and mitigating the effects of noise interference. DRL, with its ability to leverage artificial neural networks for solving high-dimensional or continuous action space problems, has been proposed as an effective method for addressing trajectory optimization challenges in UAV communication [27].

1.2. Motivation

Although reinforcement learning-based intelligent agents bring UAVs adequate adaptive machine intelligence. Applying RL in scenarios characterized by energy insufficiency where return-to-base operations are prioritized and sparse distributions of SNs in remote areas presents substantial challenges. First, the use of discrete-time models and binary reward functions often leads to sparse rewards, making it difficult for the agent to learn effectively. Second, insufficient energy causes many initial training episodes to fail in completing return-to-base operations, resulting in low sample efficiency during training. Furthermore, in training adaptive return-to-base strategies, the absence of reliable metrics to clearly differentiate between outbound and return stages creates ambiguity in determining the optimal solution. In summary, sparse reward and energy constraints significantly hinder the convergence of RL training.

There are some ways to deal with these challenges. While reward sparsity can be mitigated through reward shaping, previous experiments [26,28] indicate that shaped rewards often hinder exploration and conflict with the optimization objectives, degrading training performance. Hindsight Experience Replay (HER), proposed by Rent et al. [28], addresses sparse rewards without requiring domain knowledge. Zheng et al. [29] employed Bayesian optimization, which improves sample efficiency. However, HER relies on the environment providing clear goal information, which is unsuitable for scenario of complex goals in our study, and Bayesian optimization introduces significant computational overhead. We would prefer a simple approach if possible.

In this study, the DRL-based trajectory design problem is investigated for UAV data collection task under insufficient battery energy with safety return constraint in remote regions. First, we construct a UAV-assisted sensor data collection system. Second, we formulate the trajectory design problem as a data collection maximization problem. The optimization goal is to jointly maximize the amount of data collected and the remaining energy after the UAV returns. Finally, we convert the problem into a Markov Decision Process (MDP) and define the state space, action and reward function. However, it is hard to avoid potential side effects in a complex reward function with reward shaping. To resolve this, S-ACL is proposed as a simple yet effective method to overcome training convergence difficulties and reduce side effects. Based on S-ACL and TD3, combined with a well-designed 3-stage training and safe action exploration, we propose SS-TD3 to solve the trajectory design problem and conducted three group of experiments to verify the performance of SS-TD3, the necessity of utilizing S-ACL, and the performance of S-ACL on gym environment respectively.

To the best of our knowledge, no existing research has addressed the comprehensive optimization of data collection tasks using reinforcement learning in scenarios characterized by insufficient energy and sparse rewards. This work seeks to fill that gap by focusing on such challenging scenarios. In addressing the key issues of this study, our primary contributions are summarized as follows.

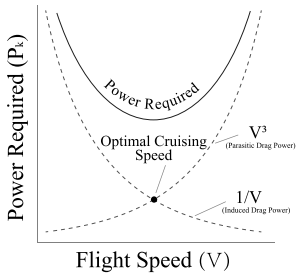

Fig. 1. Typical power required curve vs speed [30].

1.3. Contribution

(1) This study addresses the problem of optimizing UAV trajectory planning for data collection tasks in continuous space, taking into account the restrictions of UAV return and insufficient battery energy. A DRL algorithm, SS-TD3, is proposed to facilitate efficient task execution and energy conservation in UAVs.
(2) S-ACL is proposed as the foundation framework for SS-TD3. This method enhances the performance of Actor–Critic-based RL algorithms by allowing greater flexibility in the design of the reward function, exploration methods, and the parameters through staged training.
(3) Time model is improved in simulation experiments by replacing the discrete-time model with a segmented continuous-time calculation, which significantly narrows the policy space, enhances exploration efficiency, and mitigates reward sparsity, improving the adaptability of DRL algorithms.

2. System model

In this work, we consider a UAV-assisted sensor data collection system. The system comprises M wireless, low-power, self-sustaining SNs, denoted as $m \in \mathcal{M} = \{m_1, m_2, \ldots, m_M\}$, which are randomly distributed within the target area of size W × L m^2. The position of a sensor node m is represented as $L_m = [x_m, y_m]$.

2.1. Time model

The UAV, labeled as k, begins its operation at the dock, with an initial position of $L_k(0) = L_{\mathrm{d}} = [x_{\mathrm{d}}, y_{\mathrm{d}}]$, travels to the target area to collect data from the sensors, and subsequently returns to the dock.

The total time for the UAV's journey from departure to return is T. At time t, where $0 \le t \le T$, the horizontal projection of the UAV's position is denoted as $L_k(t) = [x_k(t), y_k(t)]$, and its battery energy is $E_k(t)$.

To facilitate decision-making during the UAV's flight, the continuous time period T is divided into N segments. The time t_n, where $n \in \mathcal{N} = \{0, 1, \ldots, N\}$, marks the end of the nth segment. For brevity, the UAV's position at the nth step is denoted as $L_k[n] = L_k(t_n)$, where $n \in \{0, 1, \ldots, N\}$.

The UAV's flight trajectory during the mission is denoted as $\mathcal{T}_{raj} = \{T_{\mathrm{raj}}[1], T_{\mathrm{raj}}[2], \ldots, T_{\mathrm{raj}}[N]\}$, for all $n > 0$, where each segment is defined as $T_{\mathrm{raj}}[n] = \overline{L_k[n-1]L_k[n]}$.

To increase the endurance and enhance energy efficiency, the UAV is programmed to fly at an optimal cruising speed v for each trajectory segment $T_{\mathrm{raj}}[n]$, which ensures minimal energy consumption [31]. The parameter α represents the horizontal direction of the UAV in the xy-plane relative to the x-axis. The relationship between the UAV's flight energy consumption and speed is depicted in Fig. 1.

When transitioning from one trajectory segment $T_{\mathrm{raj}}[n]$ to the next $T_{\mathrm{raj}}[n+1] (\forall n, 0 < n < N)$, the UAV executes a uniformly accelerated circular turn with a fixed angular velocity ω. For simplicity, the duration of the turning maneuver is considered negligible compared to the overall flight time. To focus on the primary research objectives, the turning trajectory is sharpened, as shown in Fig. 2.

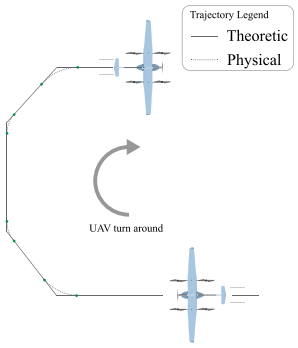

Fig. 2. Sharpen UAV steering trajectory.

The horizontal direction $\alpha[n]$ is defined as $\alpha(t_n)$, with the initialization $\alpha[0] = \alpha[1]$. The UAV's velocity for trajectory segment $T_{\mathrm{raj}}[n]$ is given by

$$V_k[n] = [v \cos \alpha[n], v \sin \alpha[n]]\,. \tag{1}$$

2.2. Communication model

The SNs are assumed to be equipped with omnidirectional antennas, while the UAV is equipped with a miniature base station and an adaptive array antenna. This configuration enables the UAV to simultaneously receive signals from multiple directions using beamforming technology. The UAV operates at a horizontal plane at a stable height H above the SNs. It is assumed that the SNs are positioned at ground level, ignoring height differences caused by environmental factors, such as SNs positioned on trees in forests or floating on the ocean surface in marine scenarios. In practice, the flight altitude H is set as the minimum height necessary to adapt to the terrain or avoid obstacles, minimizing the need for frequent altitude changes. Although this goes beyond our current consideration, we recognize that for environments with significant obstacles or complex terrain, or when dealing with particularly large amounts of data, the approach of fixing the UAV altitude plane may not be adequate. This could miss the opportunity to significantly improve communication conditions while spending relatively less energy to do maneuvering, thus resulting a suboptimal solution.

Due to the low data generation rate of small SNs, the accumulated data volume w_m of each SN is considered constant during the UAV's mission, where $0 \le w_m \le w_{\max}, \forall m \in \mathcal{M}$.

The UAV can activate SNs and establish network connections from a considerable distance. However, to ensure adequate communication quality, data transmission occurs only when the distance between the UAV's projection point and an SN is less than R. The set of SNs meeting this condition at time t is denoted as $\mathcal{M}'(t) \subseteq \mathcal{M}$, satisfying

$$d_m(t) = |L_k(t) - L_m| \le R, m \in \mathcal{M}'(t). \tag{2}$$

Furthermore, since UAV flight at a relative low speed, the Doppler effect caused by the UAV mobility is assumed to be well estimated and then compensated at the receiver [32,33]. According to actual experimental reports, UAVs can establish line-of-sight (LoS) links with ground SNs at sufficiently high altitudes, and typically experience rich scattering as well as small-scale fading [34,35]. Along the UAV trajectory, the LoS probability in a local region generally is not identical to that averaged over the whole area of interest. Therefore, we formulate channel as Rician fading model. Since time-varying elevation angle caused by UAV mobility exponentially affect the reflection, scattering, and obstruction [36], such an elevation angle-dependent Rician fading model is more practical than the conventional simplified LoS model. The channel between UAV and SN m is described as $|h_m(t)|^2 = \sqrt{\beta_m(t)}\cdot$

$g_m(t)$ [37], where $\beta_m(t) = \beta_0 d_m^{-a}(t)$ is the large-scale average channel power gain that takes into account signal attenuation, including path loss and shadowing. a denote the path loss exponent. β_0 is the average channel power gain at the reference distance $d_0 = 1\ m$. The small-scale fading coefficient is adopted as follows

$$g_m(t) = \sqrt{\frac{K_m(t)}{K_m(t)+1}}\eth + \sqrt{\frac{1}{K_m(t)+1}}\tilde{\eth}, \tag{3}$$

where $\eth$ and $\tilde{\eth}$ are the deterministic LoS channel component with $|\eth| = 1$ and the random scattering component, which is a zero-mean unit variance Circularly Symmetric Complex Gaussian (CSCG) random variable, respectively. Among them, $K_m(t) = A_1 \exp(A_2\theta_m(t))$ denote the Rician factor [37], where $\theta_m(t)$ is the elevation angle given by $\theta_m(t) = \arctan(\frac{H}{d_m(t)})$, and A_1, A_2 are constant coefficients depending on the Rician factor $K_m(t)$ of specific environment, which satisfies $K_{min} \le K_m(t) \le K_{max}$ [38]. P_{tran} is the transmission power of SN. σ^2 is the received noise power, and the total bandwidth is B. At the same moment t, orthogonal frequency division multiplexing technology is applied on UAVs for $J(t)$ SNs to evenly distribute bandwidth to provide data collection services. Considering the data transmission of UAV SN m at time t, the maximum achievable rate is

$$r_m(t) = \frac{B}{J(t)} \log_2\left(1 + \frac{P_{tran}|h_m(t)|^2}{\sigma^2}\right). \tag{4}$$

On trajectory $T_{traj}[n]$, the amount of data collected by the UAV is represented as

$$c[n] = \sum_{m\in M'} min\left\{\sum_{j\in\mathcal{J}[n]}\int_0^{t_{m,j}[n]} r_m(t)\mathrm{d}t,\ w_m\right\}, \tag{5}$$

where $t_{m,j}[n]$ indicates the maximum communication duration while the UAV is communicating with sensor node m and the number of shared bandwidth satisfies $J(t) = j$, and $\mathcal{J}[n] = \{J(t) \mid t_{n-1} < t < t_n\}$ represents the time data from $t[n-1]$ to $t[n]$. Formal expressions can be perplexing, but in practice, programming implementation is relatively easy. We record the process required to calculate $t_{m,j}[n]$ and $J(t)$ in Appendix A.

2.3. Energy model

The initial energy of the UAV is $\hat{\mathrm{E}}_k$. During flight, the UAV must overcome air resistance and gravity, and additional thrust is required for acceleration. The instantaneous power consumption for propulsion [39] is given by

$$P_k(t) = \gamma_1 \|V(t)\|^3 + \frac{\gamma_2}{\|V(t)\|}\left(1 + \frac{\|\psi(t)\|^2}{g^2}\right), \tag{6}$$

where $\psi(t)$ denote the centrifugal acceleration of the UAV, g denote the gravitational acceleration with a nominal value of 9.8 $\mathrm{m/s^2}$, and γ_1, γ_2 are internal fixed parameters such as the weight of the UAV, wing area, and air density [40,41]. For the trajectory $T_{traj}[n]$, the UAV turning time is: $t^{\psi} = (\alpha[n] - \alpha[n-1])\omega^{-1}$. The energy consumption of the UAV along the flight trajectory is expressed as:

$$E_t[n] = P_k\left(\frac{\|T_{\mathrm{raj}}[n]\|}{v} - t^{\psi}\right)(t_{n-1} - t^{\psi}) + t^{\psi} P_k t_{n-1}. \tag{7}$$

The UAV uses a fixed angular velocity for turning, as t_{n-1} represents the moment when the UAV is considered to have completed a uniformly accelerated turn. Since the speed remains v, during the turning time t^{ψ}, the tangential acceleration $a_{\parallel} = 0$, $a_{\perp} = \bar{a}$, thus the average acceleration is given as follows

$$\begin{aligned}\|\psi(n)\| &= \frac{1}{t^{\psi}}\|V[n] - V[n-1]\| \\ &= \frac{v}{t^{\psi}}\sqrt{2 - 2\cos(\alpha[n] - \alpha[n-1])}.\end{aligned} \tag{8}$$

Basing on Eq. (8), the minimum reserved return energy consumption is calculated based on the current position of the UAV, which is the energy consumed for a straight-line return to the dock:

$$\eta(t) = \frac{L_k(t) - L_{\mathrm{d}}}{v}\left(\gamma_1 v^3 + \gamma_2 v^{-1}\right). \tag{9}$$

We consider that the necessary condition for the UAV to safely return is $E_k(t) \ge \eta(t) + \tau$, where τ is the constant for the measurement error of the bottom battery energy and the additional reserved energy consumption for emergency landing. Considering that the communication data volume is not large and the communication energy consumption is relatively small compared to the UAV propulsion power consumption, ignoring communication energy consumption does not affect the optimization goal of jointly optimize data collection and energy saving. Therefore, the remaining energy consumption at the end is

$$E_k(T) = E_k(t_N) = \hat{\mathrm{E}}_k - \sum_{n\in\mathcal{N}} E_t[n]. \tag{10}$$

2.4. Problem formulation

Since UAV can recharge after returning to the dock, consuming full battery energy has little meaning, except for increasing the risk of accidents and reducing battery life. Our objective is to jointly optimize data collection and remaining energy while ensuring the UAV return to the dock safely. This is formulated as a weighted maximization problem

$$\max_{x_k, y_k} \quad \lambda_1 \sum_{n=1}^{N} c[n] + \lambda_2 E_k(t_N) \tag{11}$$

$$s.t. \quad E_k(t) \ge \eta(t) + \tau, \quad 0 \le t \le \mathrm{T} \tag{11a}$$

$$L_k(0) = P_{\mathrm{d}} \tag{11b}$$

$$\|L_k[N] - P_{\mathrm{d}}\| \le \epsilon \tag{11c}$$

$$\sum_{n\in\mathcal{N}} E_t[n] \le \hat{\mathrm{E}}_k - \tau \tag{11d}$$

$$0 \le x_k(t) \le \mathrm{W}, \quad 0 \le t \le \mathrm{T} \tag{11e}$$

$$0 \le y_k(t) \le \mathrm{L}, \quad 0 \le t \le \mathrm{T} \tag{11f}$$

where (11a) denotes the UAV's energy must always exceed the reserved energy level $\eta(t) + \tau$ to ensure safe return. (11b) denotes UAV's initial position is fixed at the docking center P_{d}. (11c) represents UAV must return to within a radius ϵ of P_{d}. (11d) represents the total energy consumed must not exceed the available energy minus the reserved energy τ.

An additional initial condition constraint ensures that the problem is situated in an energy-limited practical application scenario is list at following

$$s.t. \quad \hat{\mathrm{E}}_k < P_k \frac{1}{v} \sum_{i<\mathrm{M},\ j=i+1}^{\tilde{M}} \|L_{m_i} - L_{m_j}\|, \tag{11g}$$

where $\tilde{M}$ is a sorted list of SNs based on their nearest-neighbor distances from the origin. This constraint implies that the UAV's initial energy is insufficient for greedy, straight-line traversal of all SNs, and this may result in the UAV not being able to collect all the data from the SNs even under the optimal trajectory, highlighting the complexity of the solution space.

The positive constants λ_1 and λ_2 balance the dimensional differences between data collection and energy conservation. Adjusting the ratio allows the optimization to prioritize either maximizing collected data or conserving energy.

To evaluate algorithm performance, we use an explicit metric that highlights the trade-off between data collection and energy efficiency. The energy efficiency of data collection is defined as

$$D_{\mathrm{effi}} = \frac{\sum_{n=1}^{N} c[n]}{\sum_{m\in\mathcal{M}} w_m} \cdot \frac{E[n] - \tau}{\hat{\mathrm{E}}_k - \tau}. \tag{12}$$

If a single UAV mission collects data from all sensor nodes (SNs) and returns to dock using all available energy, the efficiency D_{effi} would reach 100%. As the denominators are constant and $\lambda_1, \lambda_2 > 0$, maximizing D_{effi} is equivalent to solving the optimization problem. In addition, we define the energy efficiency of flight distance as:

$$F_{\text{effi}} = \frac{(\gamma_1 v^3 + \gamma_2 v^{-1}) f_{\text{dist}}(t_n)}{v(\hat{\text{E}}_k - E[n])}, \tag{13}$$

where $f_{\text{dist}}(t_n)$ denotes the total flight distance during the UAV's mission. In our scenario, $F_{\text{effi}} = 100\%$ corresponds to a UAV flying at a constant velocity along a straight trajectory without acceleration, with all consumed energy used solely for steady-state flight.

Noting that constraint (11d) represents a long-term cumulative variable linked to the UAV's trajectory, while constraints (11b) and (11c) impose restrictions on the UAV's starting and ending positions, plus (11g) causing uncertainty to flight duration T. To determine the UAV's position at each step, the entire flight process must be accounted. Additionally, constraint (11g) implicitly requires solving for a binary variable set to identify SNs that are skipped. The optimal solution is therefore dependent on the spatial distribution of the SNs. The nonlinearity of constraint (11a) makes the problem non-smooth and non-differentiable, posing further challenges for traditional optimization algorithms.

Furthermore, solving the problem involves large-scale multivariable optimization, which increases the risk of the search process becoming trapped in local minima. This complexity makes traditional algorithms unsuitable for addressing dynamic optimization problems like (11). DRL, on the other hand, has proven to be an effective tool for solving complex control problems in high-dimensional continuous spaces [27]. To address the challenges of this problem, we propose SS-TD3, a DRL-based approach tailored for this scenario.

3. MDP formulation and design overview

Since the UAV's position at step $n+1$ depends only on the states of SNs and itself at step n, the UAV can be modeled as an agent, and the next target flight position is applied as action to its current state, triggering the environment to transition into the next state. As a result, the flight process of UAV can be treated as MDP with N steps. However, designing an effective reward function to facilitate DRL training poses significant challenges due to the problem's specific constraints. This section reformulates the coupled constrained problem (11) as an MDP, addressing the reward function design challenges and progressively introducing the adopted reward function while explaining the motivation for incorporating S-ACL.

The MDP is represented as $< \mathcal{S}, \mathcal{A}, \mathcal{R}, \mathcal{P}, C >$, where $\mathcal{S}$ and $\mathcal{A}$ denote the state and action spaces, $\mathcal{P}$ represents the state transition probability matrix, and $\mathcal{R}$ is the reward function. These elements are defined as follows

(a) State: At the nth time step, the state s_n comprises the following:

(1) The UAV's 2D coordinates within the operating region: $L_k[n] = [x_k(t_n), y_k(t_n)]$.
(2) The relative coordinate offset of the UAV from the dock: $L_{k,\text{d}}[n] = L_{\text{d}} - L_k[n]$.
(3) The UAV's remaining usable energy: $E'_k(t_n) = E_k(t_n) - \eta(t_n) - \tau$, excluding reserved energy for return and emergency use.
(4) Task completion rate: $\psi(t_n) = C[n](\sum_{m\in\mathcal{M}} \hat{\omega}_m)^{-1}$, where $\hat{\omega}_m$ is the initial data size of SN m.
(5) The collectible data volumes of SNs: $W(t_n) = \{\omega_m(t_n) \mid \forall m \in \mathcal{M}\}$.
(6) The coordinate offsets of SNs relative to the UAV: $L_{m,k}[n] = \{L_m - L_k[n] \mid \forall m \in \mathcal{M}\}$.

Thus, the complete state at the nth time step is represented as $s_n = \{L_k[n], L_{k,\text{d}}[n], E'_k(t_n), \psi(t_n), W(t_n), L_{m,k}[n]\}$.

(b) Action: To mitigate nonlinear complexities, the action a_n is derived from $\arctan\left(\frac{d_y}{d_x}\right)$, other than the direction indicator $\alpha[n] \in (0, 2\pi)$. Instead, it includes:

(1) Horizontal flight component: $d_x[n] \in [-1, 1]$.
(2) Vertical flight component: $d_y[n] \in [-1, 1]$.
(3) Flight duration in the current direction: $t_f[n] \in (0, 1]$, which will be scaled appropriately based on $\min\{\text{W}, \text{L}\}$.

(c) Reward: The reward function plays a critical role in applying DRL algorithms. To streamline the discussion, we define the relevant rewarding factor and auxiliary functions as follows:

(1) $\mu_c \in (0, 1]$: Rewarding for collecting data.
(2) $\mu_e \in (0, 1]$: Rewarding for remaining energy when return.
(3) $\mu_u \in (0, 1]$: Rewarding for consuming energy.
(4) $p_e > 0$: Penalty for violating energy constraint.

The less-than-or-equal comparison function, which defined as:

$$\text{Low}(x, y) = \begin{cases} 1, & \text{if } x \le y, \\ 0, & \text{otherwise.} \end{cases}$$

Additionally, the function $\text{In}(L)$ determines whether a location L falls within the UAV's forward exploration field of view Ω. By computing the boundaries of this field and substituting L into the respective inequality constraints, its status can be established. For brevity, implementation details are omitted here:

$$\text{In}(L) = \begin{cases} 1, & \text{if location } L \text{ lies within the field of view } \Omega \\ 0, & \text{otherwise.} \end{cases}$$

The goal of the reward design is to maximize data collection while minimizing energy consumption, and the initial binary reward function is constructed as follows

$$\begin{aligned} R_3(n) = {} & \mu_c c[n] \\ & + \mu_e E_k(t_n) \cdot \text{Low}(\|L_{k,d}[N]\|, \epsilon) \\ & - p_e(1 - \phi(t_n)) \cdot \text{Low}(E_k(t_n), \eta(t_n)). \end{aligned} \tag{14}$$

The reward function $R_3(n)$ consist of three parts: a reward for data collection, a reward for remaining energy when return constraints are met and a penalty for energy constraint violations.

Nevertheless, in the specific context of this work, the second part leads to a local optimum solution. Besides, $R_3(n)$ suffers from reward sparsity, leading to inefficient UAV exploration, such as hovering in areas without active SNs. This inefficiency significantly reduces sample efficiency and prolongs training times. Practical experiments reveal challenges in convergence in Section 5.3. Due to above reason, we applied reward shaping and proposed an improved reward function

$$\begin{aligned} R_4(n) = {} & R_3(n) \\ & + \mu_u(\hat{\text{E}}_k - E_k(t_n)) \cdot \text{Low}(\|L_{k,d}[N]\|, \epsilon) \\ & + \sum_{m\in\mathcal{M}} \text{In}(L_m)\mu_c \omega_m(\|L_{m,k}[n]\| + R)^{-1} \\ & + \text{In}(L_{\text{d}}) \sqrt{\frac{\max\{\|L_{k,d}[n]\| - \|L_{k,d}[n-1]\|, 0\}}{\|L_{k,d}[n]\| + \|T_{\text{raj}}[n]\|(t_n - t_{n-1})^{-1}}}. \end{aligned} \tag{15}$$

Basing on $R_3(n)$, three new parts was introduced to enhance $R_4(n)$. The first part encourages UAV to consume energy. Following two parts incentive UAV to move toward SN nodes and approaching the dock more rapidly. These improvements enhancing data collection and ensuring successful returns. However, performance remains suboptimal, as the experiment in Section 5.3 shows.

Further analysis revealed conflicts among reward components due to energy constraints (11g). Maximizing data collection conflicts with the energy required for returning, leading to suboptimal exploration. Adjusting reward factors, such as μ_e and p_e, biases the UAV toward returns, reducing exploration. Neither removing penalties nor using constant return rewards resolved these issues.

To address this, we decompose the reward function into three progressive components and propose a staged training method to utilize them. This approach innovatively smooths policy transitions and improves convergence. The method presented in Section 4 integrates

$R_1(n)$, $R_2(n)$ and $R_3(n)$ with the TD3 algorithm to achieve effective results.

$$\begin{aligned} R_1(n) = R_3(n) \\ + \mu_u(\hat{\mathrm{E}}_k - E_k(t_n)) \cdot \mathrm{Low}(\|L_{k,d}[N]\|, \epsilon) \\ - \mu_e E_k(t_n) \cdot \mathrm{Low}(\|L_{k,d}[N]\|, \epsilon). \end{aligned} \tag{16}$$

Compared to $R_3(n)$, the reward $R_1(n)$ of 1st stage removes the remaining energy reward from returns, prioritizing exploration over energy conservation. This lays the foundation for subsequent energy-saving optimization.

$$\begin{aligned} R_2(n) = R_3(n) \\ + \mu_u(\hat{\mathrm{E}}_k - E_k(n)) \cdot \mathrm{Low}(\|L_{k,d}[N]\|, \epsilon). \end{aligned} \tag{17}$$

The reward $R_2(n)$ of 2nd stage eliminates guiding rewards to reduce side effects. With $R_1(n)$ as a foundation, $R_2(n)$ encourage saving battery energy and improves UAV performance.

Finally, the simplified binary reward $R_3(n)$ balances data collection and return priorities, while reducing side effects, which achieves effective training and performance.

Algorithm 1 S-ACL Algorithm

Input: $K, \varphi_k, \rho_k, R_k(s,a), \tilde{A}_k, N_k, \gamma_k, \alpha_k$, Actor network π_θ, Critic network Q_ϕ

for $k = 1$ to K **do**
  Apply Xavier distribution on θ_k, ϕ_k
  Clear replay buffer R
  if $k > 1$ **then**
    $\theta_{k-1}, \phi_{k-1} \leftarrow \mathrm{Load}(f_{\theta,\phi})$
    $\phi_k \leftarrow (1-\rho_k)\phi_k + \rho_k \phi_{k-1}$
    $\theta_k \leftarrow \theta_{k-1}$
  end if
  Initialize environment with reward $R_k(s,a)$ as Env
  for $e = 1$ to N_k **do**
    $s \leftarrow s_0$ Obtain initial state by resetting Env
    Set count $n \leftarrow 0$, terminated flag $D \leftarrow 0$
    repeat
      Select action $a \leftarrow A_k(\pi_\theta, e, \dots)$
      Interact $(s', a, r, d) \leftarrow Env(a)$
      Store transition (s, a, r, d) into R
      $s \leftarrow s'$, $D \leftarrow d$
      if update available **then**
        Sample mini-batch as $(\bar{s}, \bar{a}, \bar{r}, \bar{s}')$ from R
        $\delta_\phi \leftarrow \mathrm{Loss}(\bar{r} + \gamma_k Q_\phi(\bar{s}', \bar{a}), Q_\phi(\bar{s}, \bar{a}))$
        $\phi \leftarrow \phi + \alpha_k \nabla_\phi(\delta_\phi)$
        $n \leftarrow n + 1$
        if $n > \varphi_k$ **then**
          $\nabla_\theta J(\theta) \leftarrow \nabla_\theta \log \pi_\theta(a_t|s_t) \cdot G(\delta_\phi, \pi_\theta, \pi_{\mathrm{old}}, s_t, a_t)$
          $\theta \leftarrow \theta + \alpha_k \nabla_\theta J(\theta)$
        end if
      end if
    until $D == 1$
  end for
  $f_{\theta,\phi} \leftarrow \mathrm{Store}(\theta_k, \phi_k)$
end for

4. Proposed scheme

In this section, we introduce the S-ACL and SS-TD3 algorithms. The SS-TD3 algorithm, designed to enhance convergence through improved exploration techniques, is applied in conjunction with the S-ACL method to maximize data collection efficiency while ensuring the UAV returns successfully.

4.1. S-ACL

To make reinforcement learning methods more effective in solving the problem outlined in this paper and to enhance agent performance, we propose the Staged Actor–Critic based reinforcement Learning (S-ACL) method, which is built upon the AC reinforcement learning framework. The pseudocode is provided in Algorithm 1. The method utilizes the Critic network to take both state and action as inputs, leveraging the strengths of AC algorithms. By maintaining a consistent state space, the parameters of the Actor network, which are responsible for generating effective policies, can be reused. This allows for incremental training in the existing solution space, thus providing flexibility in defining the reward function and exploration strategy. Specifically, the complex reward function can be splitted according to a progressive objective, combining different training parameters and exploration strategies in multiple stages (see Fig. 3).

Algorithm 2 SS-TD3 Algorithm

Input: $\varphi_k, \rho_k, R_1(n), R_2(n), R_3(n), \gamma_k, \alpha_k$, Gaussian noise σ_k, $k \in \{1,2,3\}$

$\tilde{A}_1$ and $\tilde{A}_2$: Random exploration with $\zeta(t)$, $\tilde{A}_3$: $\epsilon - Greedy$ with exploration probability $\epsilon(t)$
for $k = 1$ to 3 **do**
  Initialize critic networks Q_{ϕ_1}, Q_{ϕ_2}, actor network π_θ
  Apply Xavier distribution on ϕ_1, ϕ_2, θ
  if $k > 1$ **then**
    $\theta', \phi_1', \phi_2' \leftarrow \mathrm{Load}(f_{\theta,\phi})$
    $\phi_i \leftarrow (1-\rho_k)\phi_i + \rho_k \phi_{i-1}'$, $i = 1,2$
    $\theta \leftarrow \theta'$
  end if
  Initialize: $Q_{\phi_1}' \leftarrow Q_{\phi_1}$, $Q_{\phi_2}' \leftarrow Q_{\phi_2}$, $\pi_\phi' \leftarrow \pi_\phi$
  Clear replay buffer R
  Initialize environment with reward $R_k(s,a)$ as Env
  for $e = 1$ to N_k **do**
    $s_0 \leftarrow$ Reset environment to obtain initial state
    Set count $n \leftarrow 0$, terminated flag $D \leftarrow 0$
    repeat
      repeat
        Select action $a(t) \leftarrow A_k(\pi_\theta, e, \dots)$
      until $\zeta(t) = 1$ and $a(t)$ satisfies (11a)
      Interact $(s', a, r, d) \leftarrow Env(a)$
      Store transition (s, a, r, d) into R
      $s \leftarrow s'$, $D \leftarrow d$
      if update available **then**
        Sample mini-batch $(\bar{s}, \bar{a}, \bar{r}, \bar{s}')$ from R
        $\hat{a} \leftarrow \pi_\phi'(s') + \sigma \cdot \mathrm{clip}(a_{low}, a_{high})$
        $y \leftarrow \bar{r} + (1-\bar{d})\gamma_k \min_{i=1,2}\{Q_{\phi_i'}(\bar{s}', \hat{a})\}$
        $\delta_\phi \leftarrow MSE(y, Q_\phi(\bar{s}, \bar{a}))$
        $\phi \leftarrow \phi + \alpha_k \nabla_\phi(\delta_\phi)$
        if $n > \varphi_k$ **then**
          $\nabla_\theta J(\theta) \leftarrow \nabla_\theta \log \pi_\theta(a_t|s_t) \cdot Q_\phi(s_t, a)\big|_{a=\pi_\theta(s_t)}$
          $\theta \leftarrow \theta + \alpha_k \nabla_\theta J(\theta)$
        end if
        $\phi_i' \leftarrow \tau\phi_i + (1-\tau)\phi_i'$, $i = 1,2$
        $\theta' \leftarrow \tau\theta + (1-\tau)\theta'$, $n \leftarrow n + 1$
      end if
    until $D == 1$
  end for
  $f_{\theta,\phi} \leftarrow \mathrm{Store}(\phi_1, \phi_2, \theta)$
end for

Moreover, since each new stage refreshes the replay buffer with experiences from the previous stage's more effective policy network, low-quality samples are filtered out, leading to more efficient sample usage and improved training performance. The S-ACL method is

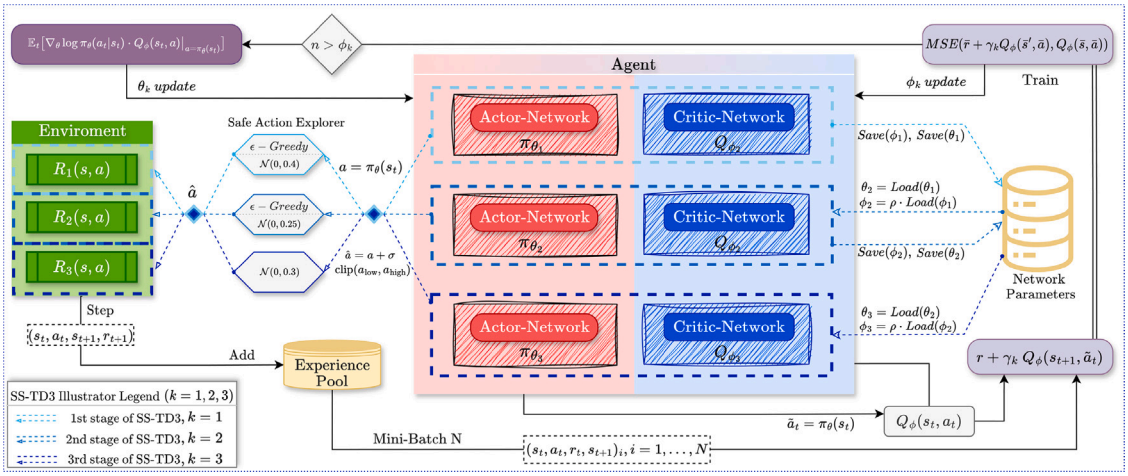

Fig. 3. SS-TD3 Illustration.

inherently suitable for a variety of AC-based reinforcement learning algorithms, such as A2C, DDPG, and PPO. By carefully selecting reward functions that are well-aligned with the problem's objectives, the original complexity is reduced, thereby enabling better alignment of the reinforcement learning agent's training goals and enhancing its overall performance.

For the S-ACL training over K stages, $k \in \mathcal{K} = 1, 2, \ldots, K$, we prepare K reward functions $R_k(s, a)$ and exploration strategies A_k, with each stage training for N_k episodes, setting the Actor network update delay φ_k, and the Critic parameter inheritance rate $\rho_k \in (0, 1)$.

Different AC type algorithms implement the update factor $G(\cdot)$ for the Critic's output in different ways. In A2C and its derivatives, this is represented as $G_{A2C} = A_{dv}(s_t, a_t)$, where $A_{dv}(s_t, a_t)$ is the advantage function. For DDPG, it is $G_{\text{DDPG}} = \nabla_a Q_\phi(s_t, a)\big|a = \pi\theta(s_t)$; for SAC, it is $G_{\text{SAC}} = \nabla_a Q(s_t, a)\big|a = \pi\theta(s_t) + \alpha\nabla_\theta \mathcal{H}(\pi_\theta)$; and for PPO, it is $G_{\text{PPO}} = \min\big(\rho \cdot A_{dv}(s_t, a_t), \text{clip}(\rho, 1-\epsilon, 1+\epsilon) \cdot A_{dv}(s_t, a_t)\big)$. In general, we express this as $G(\delta_\phi, \pi_\theta, \pi_{\text{old}}, s_t, a_t)$, and the original algorithm implementations should be followed during use. Due to this non-invasive nature, S-ACL provides an easy-to-implement training method.

The Actor network training delay typically accounts for about 10% of the total training time for each stage. The parameter φ_k should be set based on practical considerations, such as when the Critic network loss is relatively stable. This parameter also serves as the pretraining time for the Critic. In stages that reuse the Actor network parameters, the proportion of randomly generated actions during interaction with the environment should not be excessively high. Instead, noise-based exploration is recommended.

The *Store* and *Load* functions refer to storing and loading model parameters from the file $f_{\theta,\phi}$. After completing training in each stage and saving the parameters, training can be paused, which is beneficial for practical use. This step is optional, and after training each stage, a copy of the Actor network is created. Before starting each new stage, Critic network parameters are reset either randomly using the *Xavier* method or by resetting $(1 - \rho_k)$ of the parameters. The entire process continues through all stages. When using the *Polyak* method to inherit part of the Critic parameters from the previous stage, setting $\rho_k = 0.3$ can help reduce the pretraining time for the Critic. However, ρ_k should not be set too high, as this can lead to difficulties in eliminating side effects.

4.2. SS-TD3

To planning the UAV trajectory for problem (11), we introduce the SS-TD3 algorithm, which combines the TD3 algorithm with the previously discussed S-ACL method.

As its name implies, SS-TD3 leverages TD3 [27], a state-of-the-art AC algorithm for continuous control tasks, as the foundation for our design. The pseudocode for SS-TD3 is presented in Algorithm 2. The Staged-Safe-action-TD3 (SS-TD3) algorithm integrates three-stage training approach from S-ACL with the TD3 reinforcement learning algorithm, incorporating a safety-focused exploration strategy.

In the first stage, we apply a relatively high Gaussian noise $\sigma_1 = \mathcal{N}(0, 0.4)$ along with an ϵ-Greedy random exploration strategy. The reward function $R_1(n)$ is selected to maximize the UAV's exploration of the target area, while simultaneously guiding it to return and approach SNs for data collection. In the second stage, the Gaussian noise is reduced to $\sigma_2 = \mathcal{N}(0, 0.25)$, and the safe exploration strategy is maintained to allow the UAV to learn the return path. The reward function $R_2(n)$ is chosen to eliminate the guiding reward and optimizes the UAV's strategy to balance exploration with battery energy conservation. In the final stage, moderate Gaussian noise $\sigma_3 = \mathcal{N}(0, 0.3)$ is applied, and noise decay is introduced to facilitate the observation of final training outcomes in the last episodes. The reward function $R_3(n)$ is applied to focus on optimizing the problem (11) during training and aims to minimize the side effects of the reward function.

By structuring the training in this way, SS-TD3 efficiently combines exploration and safety, allowing for optimal trajectory planning and energy efficiency in UAV operations.

5. Experiments

To evaluate the performance of the SS-TD3 algorithm, we conducted a comparative experiments involving the proposed SS-TD3 and implemented baseline methods, listed as below. Further, Section 5.3 and Section 5.4 conducted another two experiments to demonstrate the necessity of applying S-ACL and the performance of S-ACL.

5.1. Simulation settings

1. **Greedy Search (GS)**: The GS algorithm is employed to determine a trajectory that guarantees a successful return, after which it attempts to enhance data collection by optimizing the insertion of discarded nodes along the path.
2. **Ant Colony Optimization (ACO)**: The classic original ACO algorithm is utilized to find the shortest path to visit SNs nodes based on pheromone trails, with no assurance of ensuring a successful return.
3. **Constraint Ant Colony Optimization (CACO)**: This algorithm is a modified version of the original ACO algorith, incorporating pheromone-based adaptations to meet energy consumption and return constraints. The ants search for the shortest path while

Table 1
Simulation parameters.

Parameters	Value
bandwidth (B) [42]	10 MHz
noise power (σ^2) [43]	−110 dBm
UAV height (H) [44]	120 m
proportion parameter (γ_1)	9.26×10^{-4}
proportion parameter (γ_2)	2250
channel gain at unit distance (β_0)	−50 dB
power efficient speed (ν)	36.72 KM/H
transmission radius (R) [44]	60 m
UAV battery energy ($\hat{E}_k$)	120,000 J
reserved battery energy (τ)	20,000 J
actor delay of 2nd stage (ϕ_2)	3×10^3
actor delay of 3rd stage (ϕ_3)	4×10^3
inherit rate of 2nd 3rd stage (ρ_2, ρ_3)	0.2
replay buffer size	1×10^5
polak factor for target network	0.002
ant numbers	20
heuristic algorithm iterations	500
pheromone decay	0.95

considering the available remaining energy. If the remaining energy is insufficient for return, the exploration of the path is terminated. The pheromone release rate is influenced by the variance in data volumes at SNs nodes, and path optimization is performed accordingly. These improvements, compared to the basic ACO algorithm, significantly enhance both the reliability of return and the efficiency of data collection.

To maintain focus on the primary research objectives, the details of the implemented baseline algorithm are presented in Appendix B.

The GS algorithm outputs a trajectory directly after searching. The heuristic algorithm provides a converged trajectory after a sufficient number of iterations. In contrast, the result of the SS-TD3 algorithm is a trained reinforcement learning model. We use the trained model to interact with the environment in the evaluation mode, recording the results of each interaction. The baseline algorithm's trajectory is converted into a series of actions, with the corresponding interaction results with the environment recorded.

During each simulation, four algorithms are executed simultaneously, and testing is conducted in 11 randomly generated environments. A total of 20 SNs are randomly generated with a minimum inter-node distance in each environment. The total data volume of SNs in each environment is constant and is randomly partitioned among all SNs using an integer random method. To ensure fairness, the initial environment is replicated for each group of algorithms. The baseline algorithm's trajectory, after sufficient convergence, and the evaluation of the SS-TD3 model are repeated 300 times across all random environments. Table 2 records the remaining battery energy, return to base (converted to 0/1 to averaged as back ratio), total flight distance and data ratio result in each group when the evaluation finish. In the end, the outputs are averaged to obtain the final results, which are presented in Fig. 8.

In the SS-TD3 algorithm, certain parameters differ across the three training stages. The learning rates for the Critic are set to: $0.001, 0.001, 0.0001$, and the discount factors are: $0.975, 0.99, 0.968$. The learning rate of the Actor is always 0.1 times that of the Critic. We adopt the network with 512×512 Multi-Layer Perceptron (MLP) architecture and ReLU activation. The number of training episodes for the three stages are set to: $8000, 6000, 5000$, and the corresponding training curves are shown in Fig. 4 and Fig. 5. For the ACO and CACO algorithms, the distance index is twice the pheromone index, and the weight index for node data volume is set to a reasonable value. All other key parameters are listed in Table 1, with the remaining parameters set to optimal values as determined by the experiments. These parameters are not further elaborated here.

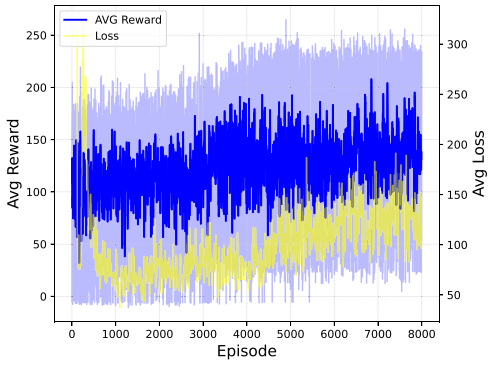

Fig. 4. 1st stage of SS-TD3.

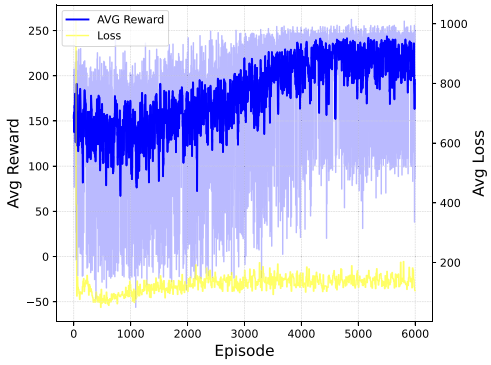

Fig. 5. 2nd stage of SS-TD3.

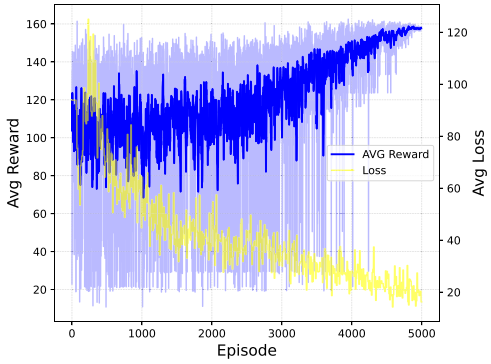

Fig. 6. 3rd stage of SS-TD3.

5.2. SS-TD3 performance

Fig. 4 and Fig. 5 present the training curves for the first and second stage of SS-TD3. The segmented time model ensures the effectiveness of the baseline exploration strategy, maintaining a certain level of exploration efficiency. Additionally, with the aid of a safe exploration strategy, random exploration can still yield some rewards. As indicated by reward $R(1)$, the majority of the initial rewards are related to the energy consumption during the return trip. This significantly motivates further exploration. While the average reward shows only a slight increase, the maximum reward reaches values exceeding 200. In the second stage of training, the underlying S-ACL method successfully inherits the strategic advantages from the first stage, leading to a further improvement in the reward and bringing the average reward closer to the peak. Due to the substantial reward loss resulting from UAV return failures, the actual reward curve depicted in light blue, exhibits considerable fluctuations during training. However, these fluctuations

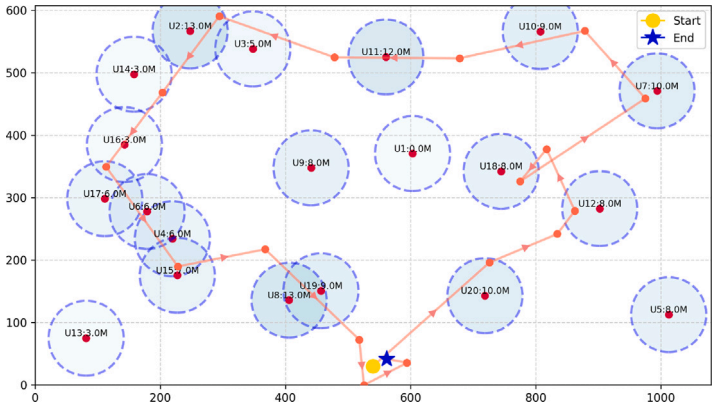

Fig. 7. Captured example trajectory at SS-TD3 3rd stage.

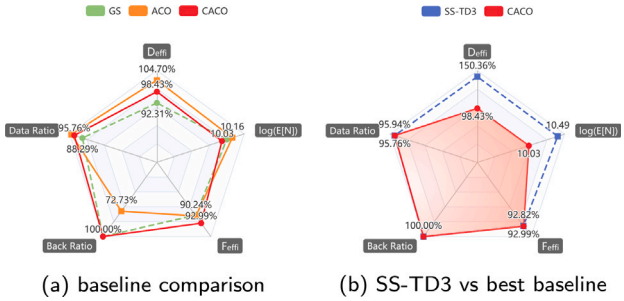

Fig. 8. SS-TD3 vs baselines.

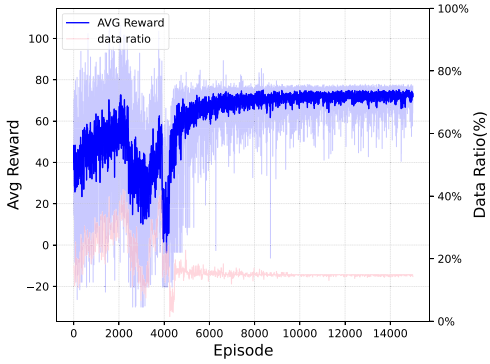

Fig. 9. S-TD3 with binary reward.

become more stable over time and gradually decrease as the training converges.

Fig. 6 illustrates the final stage of training, where the Actor network's initial policy already shows strong performance. As the noise and safe random exploration strategies diminish, the mean of the reward curve begins to rise and ultimately reaches its optimal value toward the end. Notably, some episodes in the early stages of training achieve exceptionally high rewards. Given the influence of initial noise on the network's actions and the role of the safe exploration strategy in ensuring successful return, it is reasonable for some episodes to yield such high rewards. As the exploration factor gradually decays to zero over time, maintaining a stable return and reward values requires the agent to rely on its own learned behaviors. The steady increase in the reward curve indicates that the agent has learned the optimal trajectory, achieving successful returns while also maintaining a high data collection rate.

Fig. 7 depicts a trajectory diagram for an episode near the end of the final stage of training in the SS-TD3 algorithm. In this study's scenario, the overhead does not stem from data transmission. Therefore, the UAV's trajectory optimization strategy should fit the positions of specific SNs using linear regression. It must also ensure that the communication time aligns with the data volume within the effective transmission range of SNs with high data loads. Without designing a specific reward function for this characteristic of the optimal trajectory, the DRL neural network automatically captures this feature through gradient optimization in high-dimensional space.

Additionally, due to the introduction of random noise during training, the UAV initially missed SN numbered 18 and subsequently deviated from its intended path. However, as illustrated in the figure, the UAV, guided by the reinforcement learning agent, adaptively corrected its trajectory. The ability to resist certain noise disturbances is one of the reasons we favor reinforcement learning for training the agent.

Fig. 8(a) illustrates that the ACO algorithm achieves the highest energy efficiency among all baselines at the maximum data collection rate. However, it fails to guarantee reliable return to the base. The data collection rate of the GS algorithm is limited to 88%. Among the baselines, the CACO algorithm emerges as the best in meeting operational requirements while maintaining balanced overall performance.

Fig. 8(b) depicts a comparison between the SS-TD3 algorithm and the CACO algorithm. Both achieve a 100% return rate and demonstrate comparable performance in terms of average data collection rate and flight energy consumption. Benefiting from its significantly higher residual energy, SS-TD3 outperforms the CACO algorithm and other baseline methods in energy efficiency.

5.3. Necessity of S-ACL

Based on the S-ACL method, SS-TD3 introduces two additional training stages. To highlight the necessity of using the S-ACL method, we illustrate the performance of the Safe-action-TD3 (S-TD3) algorithm, which does not incorporate S-ACL, within the environment Env.01. We record the data ratio in training along with the gained reward. Training runs on both binary reward function $R_3(n)$ and complex reward function $R_4(n)$ with reward shaping. Both variants were trained for 15,000 episodes, with the training results presented in Fig. 9 and Fig. 10.

Fig. 9 illustrates that with sufficient training episodes, the average reward curve for the binary reward function $R_3(n)$ stabilizes around 70, with the maximum stable reward not exceeding 80. The average data collection rate was below 20%, and the training performance was nearly unsuccessful. We infer that more than half of the rewards stem from remaining energy consumption, which led the agent to forgo the pursuit of data collection rewards and become trapped in a local optimum. Additionally, the sparsity of the binary reward function exacerbates this issue.

Fig. 10 demonstrates that with adequate training, the reward function $R_4(n)$ yields an average data collection rate of approximately 80%,

Table 2
Numerical results of SS-TD3,GS,ACO,CACO.

	$E[N]\,(J)\uparrow$				Successful return				$f_{dist}(t_n)\downarrow$				Data ratio↑			
	SS-TD3	GS	ACO	CACO	SS-TD3	GS	ACO	CACO	SS-TD3	GS	ACO	CACO	SS-TD3	GS	ACO	CACO
Env.01	33836.5	20569.0	23061.0	22168.9	true	true	false	true	2531.7	2893.0	2750.1	2971.4	0.925	0.693	0.829	0.884
Env.02	36581.2	24056.6	31058.7	23487.1	true	true	true	true	2490.9	2775.9	2619.4	2911.2	1.0	1.0	1.0	0.930
Env.03	30224.0	26552.1	28345.3	22409.6	true	true	true	true	2617.4	2633.9	2625.7	2915.3	0.981	0.906	1.0	0.968
Env.04	34996.5	20984.6	24206.9	20754.1	true	true	true	true	2497.7	2767.5	2825.9	2946.0	0.941	0.825	1.0	0.941
Env.05	39440.1	23821.3	21835.8	21128.5	true	true	false	true	2422.5	2891.9	2759.2	2945.2	0.979	0.802	1.0	0.987
Env.06	32935.5	31040.1	20282.2	20676.2	true	true	false	true	2542.6	2436.0	2768.4	2958.6	0.947	0.967	1.0	0.967
Env.07	37316.0	22677.3	28799.3	20519.8	true	true	true	true	2507.9	2754.4	2672.9	2856.6	0.917	1.0	1.0	0.974
Env.08	39664.9	20289.1	30919.6	29557.4	true	true	true	true	2454.9	2907.9	2658.6	2773.0	0.974	0.924	1.0	0.974
Env.09	37722.2	22877.8	23183.7	21548.8	true	true	true	true	2402.6	2870.6	2758.0	2784.0	0.967	1.0	1.0	0.967
Env.10	37513.2	32109.0	28011.9	23163.3	true	true	true	true	2412.6	2505.7	2662.5	2936.8	0.968	0.800	1.0	0.968
Env.11	36501.0	20457.1	24025.6	23596.1	true	true	true	true	2553.9	2908.4	2812.5	2860.7	0.949	0.791	1.0	0.968

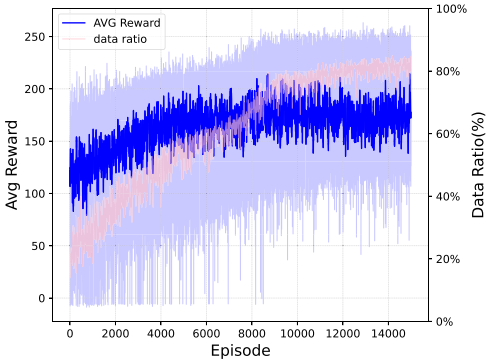

Fig. 10. S-TD3 with reward shaping.

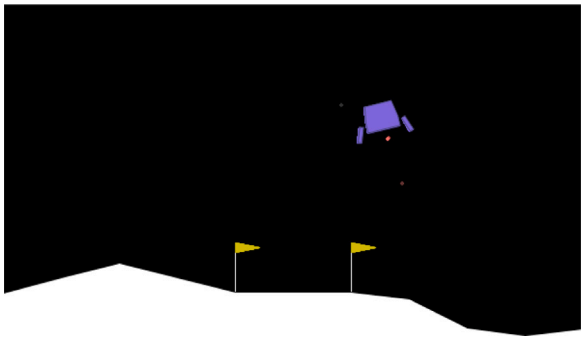

Fig. 11. Gym LunarLander-v3 Environment Illustrator.

and with further training episodes, the peak reward can reach 250, which is higher than the results from the first stage of SS-TD3. Based on our domain knowledge, using $R_4(n)$ provides a significant performance improvement over $R_3(n)$, but it still falls short of the average data collection rate of 95.94% achieved by SS-TD3 with three-stage training. As discussed in Section 3, the design of shaped rewards can sometimes conflict with the final objective and introduce unintended side effects. Although the complex function with shaped reward we developed has undergone several experimental refinements and is quite effective, designing a perfect reward function remains challenging. Such side effects are difficult to avoid in complex environments, and achieving an ideal reward function is generally a challenging task.

5.4. S-ACL performance

Further, we verify the effectiveness of the S-ACL method using the external standard environment LunarLander-v3 from the OpenAI Gym library,[3] as shown in Fig. 11. The goal of the LunarLander-v3 environment is to control a lander with jet engine switches and directional controls to achieve a safe soft landing using its two landing legs. This environment is a classic rocket trajectory optimization problem. We choose it due to its sufficient complexity which shares many similarities with our environment. The requirements for a soft landing and return at the final step are comparable, with success and failure being rewarded in completely different ways. Activating the jet engines consumes energy, and saving energy leads to higher scores. However, this environment does not impose a total energy limit, and there is no optimization objective along the trajectory, making agent training easier than in the data collection environment studied in this paper. This convenience makes it ideal for testing the S-ACL method. It should be noted, however, that the S-ACL method will perform well in more complex environments.

The Gym documentation[4] states that an episode is considered successfully solved if the total reward exceeds 200. The detailed rewards consist of the items listed below.

1. ± as the lander is closer/further to the landing pad.
2. ± as the lander is moving slower/faster.
3. is decreased the more the lander is tilted.
4. +10 for each leg that is in contact with the ground.
5. −0.03 each frame a side engine is firing.
6. −0.3 each frame the main engine is firing.
7. -100/+100 for crashing/landing safely.

We grouped experiments based on different discount factors and trained the original DDPG algorithm for timesteps=6e5. DDPG, based on the AC framework, is an effective method for solving continuous control tasks [45]. Compared to TD3, DDPG has a simpler structure and faster training speed, making it the preferred choice for testing the effectiveness of the S-ACL algorithm in the LunarLander-v3 environment.

After basic tuning of other hyperparameters, we explored the optimal discount factor. To eliminate human error, all groups were trained in identical environments with the same random seed, differing only in the discount factor. Training was recorded by completed episodes, with the cumulative reward per episode representing the episode's return. Returns and average losses were recorded, and the results were plotted in Fig. 12.

In Fig. 12, different discount factors have a significant impact on training performance. For gamma=0.97 and gamma=0.98, the agent's learning appears relatively stable. However, while gamma=0.97 yields only a few episodes with rewards exceeding 200, the majority of results

[3] The source code of lunarlander environment we used for experiments(v1.0.0 release): https://github.com/Farama-Foundation/Gymnasium/blob/196625488fc3fafef74cd97ace58ef625a41ad8c/gymnasium/envs/box2d/lunar_lander.py

[4] Document online: https://gymnasium.farama.org/v1.0.0/environments/box2d/lunar_lander/

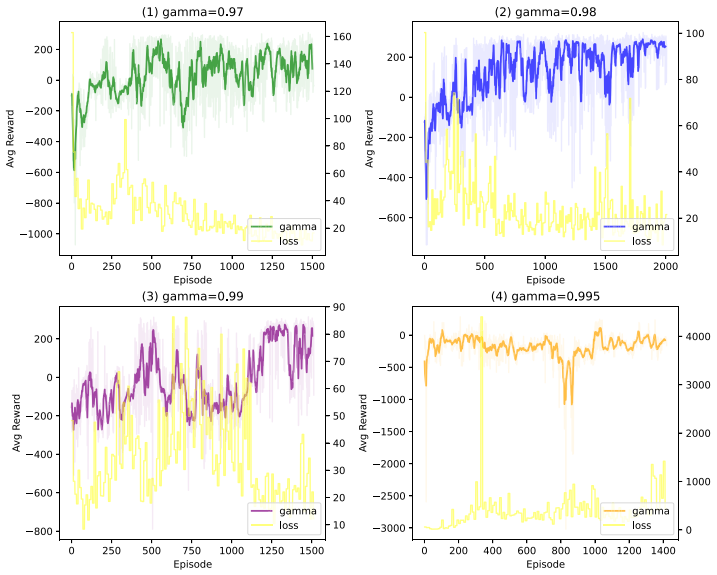

Fig. 12. DDPG training under different discount factor **gamma**.

are unsatisfactory. In contrast, gamma=0.98 achieves better performance in most episodes. The loss for gamma=0.97 remains stable but corresponds to mediocre overall training outcomes. For gamma=0.99, the loss oscillates significantly, resulting in unstable training performance within the given timesteps. Further increasing the discount factor to gamma=0.995 leads to enormous losses and nearly complete training failure.

The large number of steps per episode in the LunarLander-v3 environment magnifies the impact of the discount factor on the agent's long-term vision. It is worth noting that the total number of episodes trained varies across different discount factors due to the Gym's protocol truncating episode over 1,000 steps. This truncation occurs when the agent over-controls the lander, causing it to hover for too long, exhaust its fuel, and fail to land, ultimately leading to poor returns.

The gamma=0.98 group completes the highest number of episodes with the same total timesteps and achieves the best average returns. Therefore, we selected the gamma=0.98 group as the baseline model for validating the performance of the S-ACL algorithm.

Next, we evaluated the performance of the S-ACL algorithm. We conducted two-stage training with DDPG, naming the combined algorithm S-DDPG. Specifically, we used the gamma=0.98 model directly as the first stage of S-DDPG. The second stage parameters were set as $\phi = 1e5$ and $\rho = 0.3$, with the discount factor modified to gamma=0.995. Other hyperparameters and training settings for the second stage remained consistent with the original DDPG group.

It is important to note that in the LunarLander-v3 environment, the first three components of the reward system act as guiding rewards, helping the agent balance the lander's posture and approach the landing point to accelerate convergence. However, these rewards can sometimes cause the lander to hover in the air for an extended period. In the second stage, we removed these guiding rewards, reducing the maximum total reward by 200 points and mitigating the side effects caused by guiding rewards. This adjustment supplemented the 2nd stage training to produce the S-DDPG model. Both the S-DDPG and original DDPG models were evaluated over 1,000 episodes with identical initial environments and random seeds, and the results are shown in Fig. 13.

The results show that S-DDPG improves the average return by approximately 10% compared to DDPG and increases the landing success rate to 97.0%. However, the average return in successful landing episodes shows minimal improvement. This is because S-DDPG primarily optimizes situations where DDPG struggles to achieve successful

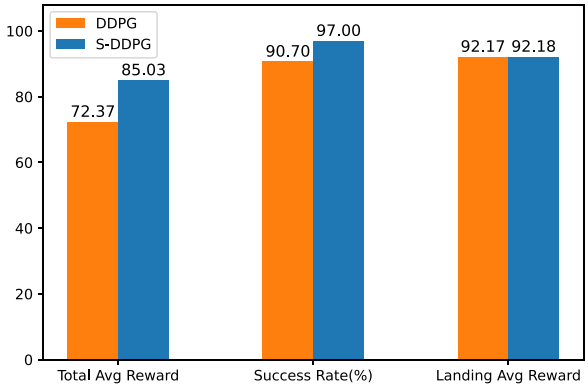

Fig. 13. DDPG vs S-DDPG on LunarLander-v3.

landings, while its energy optimization effects are less evident for already successful landing scenarios. Observations indicate that episodes with difficult landings often involve significant initial velocity offsets and may exceed 500 steps. Even when landing succeeds, it requires considerable energy consumption, resulting in low or even negative returns. For situations where landing is straightforward, S-DDPG reduces the frequency of engine activation, optimizing energy consumption; otherwise, the average return for successful landings would have decreased. Overall, it can be concluded that S-DDPG improves landing success rates without compromising landing performance.

The step count per episode in the LunarLander-v3 environment can reach 1,000. By combining the moderate discount factor gamma=0.98 and the long-term discount factor gamma=0.995 in two-stage training and leveraging reward shaping for initial training, the results demonstrate that the S-ACL method significantly enhances performance. This two-stage training approach balances the agent's long-term vision while avoiding issues such as the severe loss oscillations or gradient explosions, seen in Fig. 12(3) and Fig. 12(4) when using large discount factors from the start. It also mitigates the side effects of reward shaping, facilitating smoother training.

6. Conclusion

In this paper, we propose SS-TD3, a DRL-based algorithm designed to generate energy-efficient flight trajectories for data collection tasks

under insufficient energy and return to base constraints. SS-TD3 aims to maximize data collection while ensuring return-to-base capability, intelligently generating the UAV's motion strategy to optimize energy use. As the fundamental training framework of SS-TD3, the S-ACL algorithm leverages the actor–critic framework by decoupling the reward function, exploration strategy, and parameter configuration. This staged decomposition simplifies training in complex adversarial environments while preserving agent performance. Numerical results show that SS-TD3 significantly outperforms baseline methods in energy efficiency. Moreover, S-ACL enhances training effectiveness. Future work will explore applying S-ACL in more realistic scenarios with practical motion UAV constraints across different sensor network scales.

CRediT authorship contribution statement

Jing Mei: Writing – original draft, Methodology, Conceptualization. **Yuejia Zhang:** Software, Methodology, Investigation. **Zhao Tong:** Writing – review & editing, Data curation. **Keqin Li:** Validation, Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported by the Education Department of Hunan Province (Grant No. LXBZZ2024086).

Appendix A. Connection time

The flight time corresponding to the chord length formed by the intersection of the UAV trajectory $T_{\text{raj}}[n]$ with the circular communication range of SN m is defined as the maximum connection time $t_m[n]$. This value serves as a direct intermediate variable for calculating $t_{m,j}[n]$.

The connection time refers to the sum of flight times over the chord when the UAV passes through the user's circular communication range.

Algorithm 3 $\mathcal{J}[t]$ at connection time

Input: $T = \{(t_{M,i}, t_{N,i}), i \in \mathcal{M}\}$. results $\mathcal{R} \leftarrow []$
$T_{\text{with\_index}} \leftarrow \{(t_{M,i}, t_{N,i}, i) \mid i \in \mathcal{M}\}$
$T_{\text{sorted}} \leftarrow \text{sort}(T_{\text{with\_index}}, \text{key} = t_{M,i})$
$\mathcal{P} \leftarrow [t_{M,i}, t_{N,i} \mid i \in \mathcal{M}]$
$\mathcal{P} \leftarrow \text{unique}(\text{sort}(\mathcal{P}))$
for all $(start, end) \in \mathcal{P}$ **do**
  counter $j \leftarrow 0$
  $\mathcal{I} \leftarrow \{\}$
  for all $\tau_i \in T_{\text{sorted}}$ **do**
    if $t_{M,i} \le$ start **and** $t_{N,i} \ge$ end **then**
      $j \leftarrow j + 1$
      add i into indices
    end if
  end for
  append $(range = (start, end), j, \mathcal{I})$ into $\mathcal{R}$
end for

Given the UAV trajectory segment L with endpoints $L_1(x_0, y_0)$ and $L_2(x_1, y_1)$, where $(x_1 - x_0) \cdot (y_1 - y_0) \neq 0$, and the communication radius R of SN m at $P_m(x_p, y_p)$, the line equation for L in the coordinate system is expressed as: $L: \quad Ax + By + C = 0$. The parametric equation of L is given as follows, where t_P is the parameter corresponding to a point P on L: $(x - x_0, y - y_0) = t \cdot \bar{n}_{AB}$, where, the coefficients are defined as $A = y_1 - y_0$, $B = -(x_1 - x_0)$, and $C = x_1 y_0 - x_0 y_1$, while the direction vector $\bar{n}_{AB}$ is: $\bar{n}_{AB} = (x_1 - x_0, y_1 - y_0)$.

The objective is to calculate the chord length within the communication range of the SN, MN, which is given by:

$$MN = \begin{cases} 0 & \text{if } d = 0 \\ (|t_M - t_N|) \cdot \|\bar{n}_{AB}\| & \text{if } d > 0. \end{cases}$$

where the perpendicular distance d from the center of the circle $P_m(x_p, y_p)$ to the line L is: $d = LP = \frac{|Ax_p + By_p + C|}{\sqrt{A^2 + B^2}}$.

The line L intersects the circle centered at $P_m(x_p, y_p)$ with radius R, forming a chord MN, where the endpoints $M(x_M, y_M)$ and $N(x_N, y_N)$ are given by:

$$(x_M, y_M) = (x', y') + \lambda(-B, A)$$
$$(x_N, y_N) = (x', y') - \lambda(-B, A),$$

where, $\lambda = \frac{\sqrt{R^2 - d^2}}{\sqrt{A^2 + B^2}}$ and (x', y') is the projection of point $P_m(x_p, y_p)$ onto the line L:

$$(x', y') = (x_p, y_p) - \frac{(Ax_p + By_p + C)}{A^2 + B^2}(A, B).$$

Thus, the maximum connection time is calculated as $t_m[n] = |MN| \cdot v^{-1}$. Additionally, the time intervals $(t_{M,m}, t_{N,m})$, for all $m \in \mathcal{M}'$, are determined, allowing for the computation of the starting connection time points $\mathcal{J}[t]$. The pseudocode for this calculation is provided in Algorithm 3.

Appendix B. Baseline algorithm

Algorithm 4 Greedy Search Algorithm (GS)

Input: Position of SNs $\mathbf{L} = \{L_m | m \in \mathcal{M}\}$, UAV total energy E_{full}
Output: Optimized route r
Calculate distance matrix $\mathbf{D}$ using $\mathbf{L}$
Initialize route $r = [0]$ (starting from dock)
Nodes to select $N_s = \{0, 1, \ldots, M\}$, nodes to discard $N_d = \emptyset$
Available energy $E = E_{\text{full}}$, flying direction $\alpha = \alpha_0$
while $N_s \neq \emptyset$ **do**
  Sort N_s by row denoting distances from the current position $D[r[-1]]$, yielding $\tilde{n}_s$
  if a node in N_s can be visited and return within available energy **then**
    Obtain the index of node $\bar{n}$ and compute energy cost E_n using α via Eq. (9)
    Update direction α and available energy $E \leftarrow E - E_n$
    Add node $\bar{n}$ to route r
    Remove node $\bar{n}$ from N_s
  else
    Remove a node $n_d \in r$ such that its sum distance to other nodes in r is maximum
    Add node n_d to N_d
  end if
end while
for node $n \in N_d$ **do**
  Try to insert n into an adjacent position in r if round-trip energy constraint is satisfied
end for

Existing studies rarely consider both node weights and return constraints simultaneously, and in this work, the flight energy consumption is related to the flight direction. It is difficult to find algorithms compatible with the existing energy consumption models. Alternatively, we designed two algorithms, GS and CACO, demonstrated in Algorithm 4 and Algorithm 5, as benchmarks to evaluate the performance of SS-TD3. The former is based on greedy routing and brute-force search,

Algorithm 5 Constraint Ant Colony Optimization (CACO)

Input: SNs positions $\mathbf{L}$, SNs data $\mathbf{W}$, energy E, initial direction θ_0, best path count n, Optimal path $\mathbf{p}^*$
Output: optimal path $\mathbf{p}^*$

Initialize all paths $\mathbf{P} \leftarrow \emptyset$
Normalize $\mathbf{L}$, $\mathbf{W}$ to range $[0, 1]$
Compute distance matrix $\mathbf{D}[i, j]$ for all i, j using $\mathbf{L}$
Initialize pheromone matrix $\mathbf{\Phi} \leftarrow \frac{1}{N}$
for each iteration $t = 1$ to T **do**
  $\mathbf{P} \leftarrow \emptyset$
  for each ant $k = 1$ to M **do**
    Initialize path $r \leftarrow []$, energy E, direction $\theta \leftarrow \theta_0$
    while E satisfies constraint (11a) **do**
      Select next node j^* based on

$$P(j|i) = \frac{\Phi[i,j]^{\alpha}(\eta[i,j])^{\beta}}{\sum_{k \notin \text{visited}} \Phi[i,k]^{\alpha}(\eta[i,k])^{\beta}}$$

      Compute $\eta[i, j] = \frac{1}{\mathbf{D}[i,j]} + \gamma \frac{\mathbf{W}[j]}{\max \mathbf{W}}$
      Calculate energy consumption e_{ij} by Eq. (7)
      $E \leftarrow E - e_{ij}$, $\theta \leftarrow \theta_{ij}$
      Append j^* to path r
    end while
    Add path r to $\mathbf{P}$
  end for
  Spread pheromone on best n paths in $\mathbf{P}$

$$\Phi[i,j] \leftarrow \Phi[i,j] + \frac{W_{\text{total}}}{\mathbf{D}[i,j]}$$

  Update $\mathbf{p}^*$ if a better path exists in $\mathbf{P}$
  Apply pheromone decay: $\mathbf{\Phi} \leftarrow \mathbf{\Phi} \cdot \rho$
end for

while the latter is inspired by the pheromone mechanism of ACO algorithms and offers superior theoretical performance.

The GS algorithm consists of two stages. In the first stage, the UAV greedily flies to the nearest unvisited SN node from the current set of unvisited SNs. If visiting the selected SN would prevent the UAV from returning, the node is abandoned and added to a discard list. The UAV then re-selects greedily until all nodes have been attempted. In the second stage, the nodes in the discard list are sorted based on the sum of their distances to all other nodes. Each node in the discard list is iterated through, and the algorithm attempts to insert the node into the UAV's current trajectory between every pair of consecutive visited nodes. For each insertion, it tests whether the UAV can still return. If feasible, the node is inserted; otherwise, it remains discarded. This process continues until the discard list is fully traversed. Note that both stages are necessary: the first stage ensures a trajectory that guarantees return, while the second stage optimizes the trajectory to maximize data collection.

In CACO algorithm, ants select nodes based on pheromone levels and the data volume differences among selectable nodes. If the next step in the route would prevent the ant from returning, the route is terminated, ensuring that the pheromones always include the path back to the starting point. During the reinforcement of pheromones on the optimal path, adjustments are made to the pheromone levels along the path based on the current route and distances. Compared to the standard ACO algorithm, these modifications significantly enhance the return probability and data collection rate.

Data availability

Data will be made available on request.

References

[1] M. Mozaffari, W. Saad, M. Bennis, Y.-H. Nam, M. Debbah, A tutorial on UAVs for wireless networks: Applications, challenges, and open problems, IEEE Commun. Surv. Tutor. 21 (3) (2019) 2334–2360, http://dx.doi.org/10.1109/COMST.2019.2902862.

[2] Y. Zeng, R. Zhang, T.J. Lim, Wireless communications with unmanned aerial vehicles: Opportunities and challenges, IEEE Commun. Mag. 54 (5) (2016) 36–42, http://dx.doi.org/10.1109/MCOM.2016.7470933.

[3] M. Erdelj, E. Natalizio, K.R. Chowdhury, I.F. Akyildiz, Help from the sky: Leveraging UAVs for disaster management, IEEE Pervasive Comput. 16 (1) (2017) 24–32, http://dx.doi.org/10.1109/MPRV.2017.11.

[4] F. Cheng, S. Zhang, Z. Li, Y. Chen, N. Zhao, F.R. Yu, V.C. Leung, UAV trajectory optimization for data offloading at the edge of multiple cells, IEEE Trans. Veh. Technol. 67 (7) (2018) 6732–6736, http://dx.doi.org/10.1109/TVT.2018.2811942.

[5] N.H. Motlagh, M. Bagaa, T. Taleb, UAV-based IoT platform: A crowd surveillance use case, IEEE Commun. Mag. 55 (2) (2017) 128–134, http://dx.doi.org/10.1109/MCOM.2017.1600587CM.

[6] L.M. Borges, F.J. Velez, A.S. Lebres, Survey on the characterization and classification of wireless sensor network applications, IEEE Commun. Surv. Tutor. 16 (4) (2014) 1860–1890, http://dx.doi.org/10.1109/COMST.2014.2320073.

[7] A.S. Ali, A.A. Al-Habob, L. Bariah, O.A. Dobre, S. Muhaidat, Deep reinforcement learning for energy-efficient data dissemination through UAV networks, IEEE Open J. Commun. Soc. (2024) http://dx.doi.org/10.1109/OJCOMS.2024.3398718.

[8] J. Li, G. Sun, S. Liang, Y. Wang, A. Wang, Multi-objective uplink data transmission optimization for edge computing in UAV-assistant mobile wireless sensor networks, J. Syst. Archit. 132 (2022) 102744, http://dx.doi.org/10.1016/j.sysarc.2022.102744.

[9] J. Guo, H. Wang, W. Liu, G. Huang, J. Gui, S. Zhang, A lightweight verifiable trust based data collection approach for sensor–cloud systems, J. Syst. Archit. 119 (2021) 102219, http://dx.doi.org/10.1016/j.sysarc.2021.102219.

[10] W. Sun, Z. Bai, J. Shi, Z. Li, DDPG-based multi-UAV trajectory optimization for wsn's data collection, in: 2023 IEEE 11th International Conference on Information, Communication and Networks, ICICN, IEEE, 2023, pp. 241–247, http://dx.doi.org/10.1109/ICICN59530.2023.10393176.

[11] Y. Fang, Z. Kuang, H. Wang, S. Lin, A. Liu, Minimizing energy consumption of collaborative deployment and task offloading in two-tier UAV edge computing networks, J. Syst. Archit. 167 (2025) 103511.

[12] M. Kang, S.-W. Jeon, Energy-efficient data aggregation and collection for multi-UAV-enabled IoT networks, IEEE Wirel. Commun. Lett. (2024) http://dx.doi.org/10.1109/LWC.2024.3355934.

[13] P. Wang, Z. Yan, G. Han, H. Yang, Y. Zhao, C. Lin, N. Wang, Q. Zhang, A2E2: Aerial-assisted energy-efficient edge sensing in intelligent public transportation systems, J. Syst. Archit. 129 (2022) 102617.

[14] L. Zhang, R. Tan, Y. Zhang, J. Peng, J. Liu, K. Li, UAV-assisted dependency-aware computation offloading in device–edge–cloud collaborative computing based on improved actor–critic DRL, J. Syst. Archit. 154 (2024) 103215.

[15] X. Fu, C. Deng, A. Guerrieri, Low-AoI data collection in integrated UAV-UGV-assisted IoT systems based on deep reinforcement learning, Comput. Netw. 259 (2025) 111044, http://dx.doi.org/10.1016/j.comnet.2025.111044.

[16] M.R. Raju, S.K. Mothku, M.K. Somesula, S. Chebrolu, Age and energy aware data collection scheme for urban flood monitoring in UAV-assisted wireless sensor networks, Ad Hoc Netw. 168 (2025) 103704, http://dx.doi.org/10.1016/j.adhoc.2024.103704.

[17] Y. Guo, S. Yin, J. Hao, Resource allocation and 3D trajectory design in wireless networks assisted by rechargeable UAV, IEEE Wirel. Commun. Lett. 8 (3) (2019) 781–784, http://dx.doi.org/10.1109/LWC.2019.2892721.

[18] M. Li, S. He, H. Li, Minimizing mission completion time of UAVs by jointly optimizing the flight and data collection trajectory in UAV-enabled WSNs, IEEE Internet Things J. 9 (15) (2022) 13498–13510, http://dx.doi.org/10.1109/JIOT.2022.3142764.

[19] H. Zhang, L. Dou, B. Xin, J. Chen, M. Gan, Y. Ding, Data collection task planning of a fixed-wing unmanned aerial vehicle in forest fire monitoring, IEEE Access 9 (2021) 109847–109864, http://dx.doi.org/10.1109/ACCESS.2021.3102317.

[20] X. Zhang, S. Zhao, Y. Wang, X. Wang, X. Song, X. Li, J. Li, 3D trajectory optimization for UAV-assisted hybrid FSO/RF network with moving obstacles, IEEE Trans. Aerosp. Electron. Syst. (2024) http://dx.doi.org/10.1109/TAES.2024.3462685.

[21] J.-R. Cao, W. Wang, N. Xu, W.-T. Su, L.-X. Xing, J.-T. Su, Robust energy efficiency optimization strategy for emergency communication based on fixed-wing UAV, J. Comput. Sci. Tech. 35 (2024) 37–57, http://dx.doi.org/10.53106/199115992024043502003.

[22] P. Ramesh, J.M.L. Jeyan, Comparative analysis of fixed-wing, rotary-wing and hybrid mini unmanned aircraft systems (UAS) from the applications perspective, INCAS Bull. 14 (1) (2022) 137–151, http://dx.doi.org/10.13111/2066-8201.2022.14.1.12.

[23] Y. Zhu, B. Yang, M. Liu, Z. Li, UAV trajectory optimization for large-scale and low-power data collection: An attention-reinforced learning scheme, IEEE Trans. Wirel. Commun. (2023) http://dx.doi.org/10.1109/TWC.2023.3304900.
[24] B. Yang, Y. Yu, X. Hao, P.L. Yeoh, J. Zhang, L. Guo, Y. Li, OH-DRL: An AoI-Guaranteed energy-efficient approach for UAV-assisted IoT data collection, IEEE Trans. Wirel. Commun. (2025) http://dx.doi.org/10.1109/TWC.2025.3545451, 1–1.
[25] C.-W. Fu, M.-L. Ku, Y.-J. Chen, T.Q. Quek, UAV trajectory, user association and power control for multi-uav enabled energy harvesting communications: offline design and online reinforcement learning, IEEE Internet Things J. (2023) http://dx.doi.org/10.1109/JIOT.2023.3325841.
[26] C. Wang, J. Wang, J. Wang, X. Zhang, Deep-reinforcement-learning-based autonomous UAV navigation with sparse rewards, IEEE Internet Things J. 7 (7) (2020) 6180–6190, http://dx.doi.org/10.1109/JIOT.2020.2973193.
[27] S. Fujimoto, H. Hoof, D. Meger, Addressing function approximation error in actor-critic methods, in: International Conference on Machine Learning, PMLR, 2018, pp. 1587–1596, http://dx.doi.org/10.48550/arXiv.1802.09477.
[28] J. Ren, Y. Zhang, Y. Zeng, Y. Lan, Data-efficient deep reinforcement learning method toward scaling continuous robotic task with sparse rewards, in: 2021 IEEE International Conference on Real-Time Computing and Robotics, RCAR, IEEE, 2021, pp. 1425–1431, http://dx.doi.org/10.1109/RCAR52367.2021.9517647.
[29] L. Zheng, Y. Li, Y. Wang, G. Bai, H. He, E. Dong, Uncertainty in Bayesian reinforcement learning for robot manipulation tasks with sparse rewards, in: 2023 IEEE International Conference on Robotics and Biomimetics, ROBIO, IEEE, 2023, pp. 1–6, http://dx.doi.org/10.1109/ROBIO58561.2023.10354785.
[30] E.M. Greitzer, Z.S. Spakovszky, I.A. Waitz, Thermodynamics & propulsion, 16. Unified, 2002, Available online at http://web.mit.edu/16.unified/www/FALL/thermodynamics/notes/notes.html.
[31] Y. Zeng, R. Zhang, Energy-efficient UAV communication with trajectory optimization, IEEE Trans. Wirel. Commun. 16 (6) (2017) 3747–3760, http://dx.doi.org/10.1109/TWC.2017.2688328.
[32] L.C. Gimenez, M.C. Cascino, M. Stefan, K.I. Pedersen, A.F. Cattoni, Mobility performance in slow- and high-speed LTE real scenarios, in: 2016 IEEE 83rd Vehicular Technology Conference, VTC Spring, 2016, pp. 1–5, http://dx.doi.org/10.1109/VTCSpring.2016.7504347.
[33] S. Benouadah, L. Mishra, N. Kaabouch, Doppler shift effect on bandwidth availability for UAV communications, in: 2024 Integrated Communications, Navigation and Surveillance Conference, ICNS, IEEE, 2024, pp. 1–6.
[34] L. Wang, B. Li, UAV-enabled reliable mobile relaying under the time-varying rician fading channel, Alex. Eng. J. 64 (2023) 771–783, http://dx.doi.org/10.1016/j.aej.2022.10.049.
[35] W. Khawaja, I. Guvenc, D.W. Matolak, U.-C. Fiebig, N. Schneckenburger, A survey of air-to-ground propagation channel modeling for unmanned aerial vehicles, IEEE Commun. Surv. Tutor. 21 (3) (2019) 2361–2391, http://dx.doi.org/10.1109/COMST.2019.2915069.
[36] S. Shimamoto, et al., Channel characterization and performance evaluation of mobile communication employing stratospheric platforms, IEICE Trans. Commun. 89 (3) (2006) 937–944, http://dx.doi.org/10.1093/ietcom/e89-b.3.937.
[37] C. You, R. Zhang, 3D trajectory optimization in rician fading for UAV-enabled data harvesting, IEEE Trans. Wirel. Commun. 18 (6) (2019) 3192–3207, http://dx.doi.org/10.1109/TWC.2019.2911939.
[38] Y. Lyu, W. Wang, P. Chen, Fixed-wing UAV based air-to-ground channel measurement and modeling at 2.7GHz in rural environment, IEEE Trans. Antennas and Propagation (2024) http://dx.doi.org/10.1109/TAP.2024.3428337, 1–1.
[39] X. Huang, X. Yang, Q. Chen, J. Zhang, Task offloading optimization for UAV-assisted fog-enabled internet of things networks, IEEE Internet Things J. 9 (2) (2022) 1082–1094, http://dx.doi.org/10.1109/JIOT.2021.3078904.
[40] S. Jeong, O. Simeone, J. Kang, Mobile edge computing via a UAV-mounted cloudlet: Optimization of bit allocation and path planning, IEEE Trans. Veh. Technol. 67 (3) (2018) 2049–2063, http://dx.doi.org/10.1109/TVT.2017.2706308.
[41] M. Xu, Z. Zhao, M. Peng, Z. Ding, T.Q.S. Quek, W. Bai, Performance analysis of computation offloading in fog-radio access networks, in: ICC 2019 - 2019 IEEE International Conference on Communications, ICC, 2019, pp. 1–6, http://dx.doi.org/10.1109/ICC.2019.8761061.
[42] Y. Wang, W. Fang, Y. Ding, N. Xiong, Computation offloading optimization for UAV-assisted mobile edge computing: a deep deterministic policy gradient approach, Wirel. Netw. 27 (4) (2021) 2991–3006, http://dx.doi.org/10.1007/s11276-021-02632-z.
[43] K. Khac Nguyen, T.Q. Duong, T. Do-Duy, H. Claussen, 3D UAV trajectory and data collection optimisation via deep reinforcement learning, 2021, http://dx.doi.org/10.1109/TCOMM.2022.3148364, ArXiv E-Prints, arXiv–2106.
[44] C. Zhao, J. Liu, M. Sheng, W. Teng, Y. Zheng, J. Li, Multi-UAV trajectory planning for energy-efficient content coverage: A decentralized learning-based approach, IEEE J. Sel. Areas Commun. 39 (10) (2021) 3193–3207, http://dx.doi.org/10.1109/JSAC.2021.3088669.
[45] T. Lillicrap, Continuous control with deep reinforcement learning, 2015, http://dx.doi.org/10.48550/arXiv.1509.02971, arXiv preprint arXiv:1509.02971.

Jing Mei received the Ph.D. degree in computer science from Hunan University, China, in 2015. She is currently an associate professor in the College of Information Science and Engineering in Hunan Normal University. Her research interests include cloud computing, fog computing and mobile edge computing, high performance computing, task scheduling and resource management, etc. She has published more than 30 research articles in international conference and journals, such as *IEEE Transactions on Computers, IEEE Transactions on Parallel and Distributed System,IEEE Transactions on Service Computing, Cluster Computing,Journal of Grid Computing, Journal of Supercomputing*.

Yuejia Zhang received the B.S. degree in computer science and technology from JiangXi University of Science and Technology, Ganzhou, China, in 2023. He is currently pursuing the M.S. degree at the College of Information Science and Engineering, Hunan Normal University, Changsha, China. His research interests focus on UAV-assisted sensor networks and deep reinforcement learning.

Zhao Tong is currently an professor with Hunan Normal University. He was a visiting scholar at the Georgia State University during 2017-2018. He has author or coauthored more than 50 papers in peer-reviewed international journals and conferences, such as *IEEE Transactions on Parallel and Distributed Systems, IEEE Transactions on Services Computing, IEEE Transactions on Network and Service Management, IEEE Transactions on Vehicular Technology*, etc. His research interests include AI computing, parallel and distributed computing systems, and resource management. He is a senior member of the IEEE and CCF.

Keqin Li is a SUNY Distinguished Professor of Computer Science with the State University of New York. He is also a National Distinguished Professor with Hunan University, China. His current research interests include cloud computing, fog computing and mobile edge computing, energy-efficient computing and communication, embedded systems and cyber–physical systems, heterogeneous computing systems, big data computing, high-performance computing, CPU–GPU hybrid and cooperative computing, computer architectures and systems, computer networking, machine learning, intelligent and soft computing. He has authored or co-authored over 850 journal articles, book chapters, and refereed conference papers, and has received several best paper awards. He holds over 70 patents announced or authorized by the Chinese National Intellectual Property Administration. He is among the world's top 5 most influential scientists in parallel and distributed computing in terms of both single-year impact and career-long impact based on a composite indicator of Scopus citation database. He has chaired many international conferences. He is currently an associate editor of the ACM Computing Surveys and the *CCF Transactions on High Performance Computing*. He has served on the editorial boards of the *IEEE Transactions on Parallel and Distributed Systems*, the *IEEE Transactions on Computers*, the *IEEE Transactions on Cloud Computing*, the *IEEE Transactions on Services Computing*, and the *IEEE Transactions on Sustainable Computing*. He is an IEEE Fellow and an AAIA Fellow. He is also a Member of Academia Europaea (Academician of the Academy of Europe).