drones

MDPI

Article

Computing Efficiency Optimization for UAV-Enabled Integrated Sensing, Computing, and Communication: A Memory-Based Deep Reinforcement Learning Approach

Honghao Qi [1,2,*] and Muqing Wu [1,2]

[1] Beijing Laboratory of Advanced Information Networks, Beijing University of Posts and Telecommunications, Beijing 100876, China; wumuqing@bupt.edu.cn
[2] Beijing Key Laboratory of Network System Architecture and Convergence, Beijing University of Posts and Telecommunications, Beijing 100876, China
* Correspondence: qihonghao@bupt.edu.cn

Highlights

What are the main findings?

- A computing efficiency maximization problem for unmanned aerial vehicle (UAV)-enabled integrated sensing, computing, and communication (ISCC) networks is formulated. The beamforming vector, CPU frequency, and UAV trajectory are jointly optimized using a memory-based deep reinforcement learning (DRL) approach.
- A long short-term memory (LSTM) and twin delayed deep deterministic policy gradient (TD3)-based trajectory design and resource allocation (LTTDRA) algorithm is proposed. The proposed LTTDRA effectively enhances the computing efficiency of UAV-enabled ISCC networks.

What are the implications of the main findings?

- This study provides solid technical support for the integrated optimization of communication, sensing, and computing resources in UAV-enabled ISCC networks.
- The proposed approach offers a robust and scalable solution to the highly-coupled resource allocation and trajectory optimization in complex UAV-enabled networks.

Abstract

Unmanned aerial vehicles (UAVs) have emerged as a promising platform for supporting integrated sensing, computing, and communication (ISCC) functionality in Internet of Things (IoT) applications. This paper investigates a UAV-enabled ISCC network, where the UAV performs radar sensing and onboard edge computing with the computational assistance of ground access points (APs). Given the limited onboard energy, ensuring energy-efficient operation of UAVs is crucial to support the long-term sustainability of network performance. In this paper, we define computing efficiency as the ratio between the total number of successfully processed computational bits and the overall UAV energy consumption, under the constraint of a required sensing threshold. To maximize this performance metric, this paper jointly optimizes the beamforming vector, the CPU frequency, and the trajectory of the UAV. This optimization problem is modeled as a Markov decision process (MDP) and solved using a deep reinforcement learning (DRL) approach based on a memory mechanism. Specifically, a long short-term memory (LSTM) and twin delayed deep deterministic policy gradient (TD3)-based trajectory design and resource allocation (LTTDRA) algorithm is proposed. LSTM units are integrated into the actor and critic to effectively capture the temporal correlations in dynamic environments, thereby enhancing policy stability and accelerating algorithm convergence. The reward function is meticulously designed to alleviate sparse-penalty effects and learn high-performance strategies

Academic Editor: Petros S. Bithas

Received: 31 December 2025
Revised: 25 February 2026
Accepted: 3 March 2026
Published: 6 March 2026

in complex environments with multiple constraints. Extensive simulations are conducted under various settings and network scenarios, and the results consistently indicate that the proposed approach substantially outperforms the baseline schemes.

Keywords: UAV; ISCC; trajectory design; resource allocation; computing efficiency; DRL

1. Introduction

With the rapid advancement of diverse Internet of Things (IoT) applications, such as vehicle networking, smart agriculture, resource exploration, equipment monitoring, and emergency disaster relief, the number of wireless devices has experienced explosive growth. Consequently, the scarcity of spectrum resources has become increasingly severe [1,2]. At the same time, to achieve ubiquitous intelligent and real-time services, the next-generation wireless networks need to provide high-precision perception and ultra-low latency computing [3–6]. To meet these ambitious requirements, the integrated sensing, computing, and communication (ISCC) paradigm has been put forward as a key architectural solution [7–9]. By incorporating mobile edge computing (MEC) capabilities into conventional integrated sensing and communication (ISAC) frameworks, ISCC realizes a triple-functional design that not only improves spectrum utilization but also augments the network's computing capability, providing greater flexibility and performance advantages over traditional dual-functional ISAC schemes [10].

Recently, traditional terrestrial wireless communication has experienced significant growth in both application scope and system complexity. Due to limitations in network capacity and coverage, terrestrial wireless communication systems are increasingly unable to guarantee high-speed and highly reliable service ubiquitously, particularly under adverse environmental conditions. In this context, integrated air–ground communication architectures have drawn growing interest [11,12]. In particular, unmanned aerial vehicles (UAVs) are capable of establishing reliable line-of-sight (LoS) communication links, thereby effectively reducing node transmission power and minimizing data packet loss rates [13,14]. Through adapting their positions over time, UAVs can be strategically deployed to carry out radar sensing, data acquisition, and edge computation tasks [15,16]. Therefore, the ISCC system empowered by UAVs provides an effective approach to expanding coverage, enhancing computational efficiency, and accommodating time-varying traffic and sensing requirements [17].

1.1. Related Works

Over the past few years, research and applications of UAVs in wireless networks have witnessed significant growth, driven by advancements in aircraft technology and regulatory frameworks [18–21]. Notably, UAV-enabled ISAC has attracted substantial interest among researchers, further underscoring this trend [22]. However, for UAV-enabled ISAC, the sensing performance and communication throughput are tightly coupled with the UAV trajectory and transmit strategy, which often leads to challenging joint design problems involving scheduling, power allocation, and trajectory planning under practical constraints such as energy budget and performance requirements. In Ref. [23], the authors investigated an ISAC network in which the UAV performs target sensing tasks for user equipments (UEs) and transmits the sensing data to the base station (BS). In this work, the radar mutual information was adopted to measure the radar detection performance, and the sensing fairness among users was fully considered. In Ref. [24], a UAV-enabled ISAC network based on rate-splitting multiple access (RSMA) and time-division multiplexing (TDM) slot structure was considered to support both sensing and communication. This work aimed to maximize the sum of the throughput and the sensing data volume. UAV trajectory, ISAC scheduling strategy, common rate allocation, and transmit power were jointly optimized in this work. In Ref. [25], the Cramer–Rao lower bound was employed to simultaneously improve communication capacity and target localization performance in a UAV-ISAC system, and a multi-stage trajectory design (MSTD) scheme was proposed to obtain a high-quality trajectory solution. In Ref. [26], the authors developed a cooperative UAV-ISAC framework in which multiple UAVs collaborate to improve sensing coverage, detection accuracy, and robustness to ground clutter interference.

In the UAV-enabled ISCC architecture, the high-capacity edge server is mounted on the UAV. Sensing data and data offloaded from ground user equipments (UEs) are processed jointly at this server, which effectively reduces both energy consumption and transmission delay [27–29]. On this basis, ISCC has recently been recognized as a promising paradigm for enhancing UAV-assisted IoT services. Compared with conventional MEC that mainly focuses on task offloading, ISCC explicitly incorporates sensing as an additional service objective and constraint, which enables the UAV to simultaneously perceive the environment, communicate with ground nodes, and execute computation tasks in a tightly coordinated manner. Such a triple-functional design is particularly attractive for infrastructure-limited scenarios, where the UAV needs to autonomously adapt its sensing and computing behaviors while maintaining reliable communication links. Therefore, exploiting the UAV's mobility to jointly orchestrate sensing, computing, and communication becomes a key enabler for achieving high system efficiency, which naturally motivates the research on the unique challenges of UAV-enabled ISCC networks.

The UAV's mobility causes the communication channels, sensing geometry, and computation workload to vary over time. This temporal variability couples trajectory design with communication, computing, and sensing resource allocation, making it challenging to balance these three functionalities in a unified framework. Despite this potential, studies on UAV-enabled ISCC remain relatively scarce. In Ref. [27], a joint design of beamforming vectors and UAV trajectory was carried out to characterize the fundamental tradeoff between computation capability and sensing beampattern gain, which was solved using semidefinite programming and the concave–convex procedure. In Ref. [28], UAV trajectory, CPU frequency, transmit power of UE, and transmit power of UAV were optimized in a unified framework to minimize the system energy consumption. To handle the resulting nonconvex problem, a multi-layer iterative algorithm was proposed. In Ref. [29], the authors jointly optimized the sensing performance and information freshness in an ISCC system through jointly optimizing the sensing scheduling, sensing times, transmit power, operating frequency, and motion parameters of the UAV. An alternating optimization (AO)-based five-stage iterative algorithm was proposed to obtain solutions.

1.2. Motivations and Contributions

Numerous challenges in UAV-enabled ISCC remain unaddressed, motivating the investigation presented in this paper. Most conventional optimization strategies for UAV-enabled wireless networks rely on convex optimization algorithms. However, practical optimization problems for UAV-enabled wireless networks are inherently non-convex, necessitating complex transformations to approximate them as convex formulations. Furthermore, convex optimization approaches typically require predefined operational policies and accurate knowledge of environmental parameters, which are often difficult to obtain in dynamic and unpredictable real-world conditions. For example, the reviewed optimization approaches for UAV-enabled ISCC networks require complete information of the system in advance [27–29]. With the ongoing advancement of artificial intelligence and unmanned systems, UAVs equipped with high-performance processors can instead learn to interact with the environment by leveraging deep reinforcement learning (DRL) methods and gradually acquire near-optimal service policies without explicit knowledge of the entire system model [30–33]. For instance, Ref. [34] proposed a Soft Actor–Critic (SAC)-based scheme that jointly optimizes beamforming, power allocation, and signal-frame scheduling in a UAV-assisted ISAC architecture, where active sensing at a ground base station (GBS) and passive sensing at a relay UAV are coordinated to enable 3D localization of a target UAV with unknown position. Although DRL is a highly promising technology, its application to performance optimization in UAV-enabled wireless networks remains challenging due to resource uncertainty, dynamic network conditions, and the high concurrency of multiple tasks. In addition, subject to the constraint of the required sensing threshold, the UAV-enabled ISCC system should not only complete as many computation tasks as possible, but also explicitly account for the UAVs' limited onboard energy to maintain energy-efficient operation [35]. It is worth noting that existing works either ignored the energy consumption of the UAV [27], or only imposed an energy-capacity constraint while neglecting the energy efficiency of providing ISCC services [29]. In light of the aforementioned issues, we investigate this work.

The contributions of this article are summarized as follows:

- **ISCC system modeling and problem formulation:** We consider a UAV-enabled ISCC architecture where a rotary-wing UAV performs radar sensing and onboard edge computing with the computational assistance of APs. On this basis, we formulate a joint optimization problem that maximizes the computing efficiency, defined as the ratio of the cumulative number of computation bits successfully processed to the total energy consumed by the UAV, subject to a required sensing threshold. The optimization variables include the UAV trajectory, the beamforming vector, and the CPU operating frequency.
- **DRL-based formulation and algorithm design:** Since the formulated problem involves highly non-convex objective functions and intricately coupled variables that challenge traditional mathematical optimization, our methodology transitions to a learning-based paradigm. In this work, we model the aforementioned problem as a Markov decision process (MDP) framework. Based on this formulation, an LSTM-TD3-based trajectory design and resource allocation (LTTDRA) algorithm is proposed. LSTM units are embedded into the actor and critic to extract temporal features from historical state–action sequences. Such temporal modeling enables the agent to better capture environment dynamics, thus improving policy stability and decision quality. For this MDP, the state space and action space are carefully designed to ensure effective representation of environmental dynamics and agent behavior. The adopted continuous action space is inherently compatible with the TD3-based algorithm. Furthermore, the reward function is meticulously designed to mitigate the sparse penalty predicament and learn high-performance strategies in complex environments with multiple constraints.
- **Performance evaluation and validation:** Extensive simulations are conducted under various system settings. The numerical results indicate that LTTDRA attains faster and more stable convergence than existing DRL-based baselines and achieves significantly higher computing efficiency than all baseline schemes.

1.3. Organization

The rest of this paper is organized as follows. Section 2 establishes the system model and formulates the problem. Section 3 presents the proposed LTTDRA algorithm. Section 4 provides the simulation results. Finally, Section 5 concludes the paper.

Notations: $|\cdot|$ denotes the absolute value or modulus. $\|\cdot\|$ denotes the Euclidean norm. $\boldsymbol{a}^H$ denotes the Hermitian transpose of $\boldsymbol{a}$. $\boldsymbol{A} \otimes \boldsymbol{B}$ denotes the Kronecker product. $\mathbb{C}^{M\times N}$ denotes the space of $M \times N$ complex matrices. $\mathcal{N}(\mu, \sigma^2)$ and $\mathcal{CN}(\mu, \sigma^2)$ denote the real and circularly symmetric complex Gaussian distributions with mean μ and variance σ^2, respectively. $\boldsymbol{I}$ and $\boldsymbol{0}$ denote an identity matrix and an all-zero matrix, respectively, with appropriate dimensions. The indicator function $\mathbf{1}\{\cdot\}$ equals 1 if the stated condition holds and 0 otherwise. $\mathbb{E}[\cdot]$ denotes the mathematical expectation.

2. System Model and Problem Formulation

The UAV-enabled ISCC network is illustrated in Figure 1. The network includes a rotary-wing UAV, along with K sensing targets (STs) and N ground access points (APs). The system is considered to operate within a square area with side length L. The UAV maintains a constant altitude H while adhering to a maximum horizontal speed limit of $V_{\max}$. A uniform planar array (UPA) with $M = M_x \times M_y$ antenna elements is integrated into the UAV. The task time is divided into T discrete slots. At each time slot t, the UAV communicates with ground APs and detects STs through the unified beams. Meanwhile, under a partial offloading policy, the UAV can execute a portion of the computation locally using its onboard processor while offloading the remaining workload to the ground APs. This cooperative architecture enables flexible task splitting and resource sharing between the UAV and multiple APs, thereby supporting efficient and scalable ISCC operations in dynamic environments.

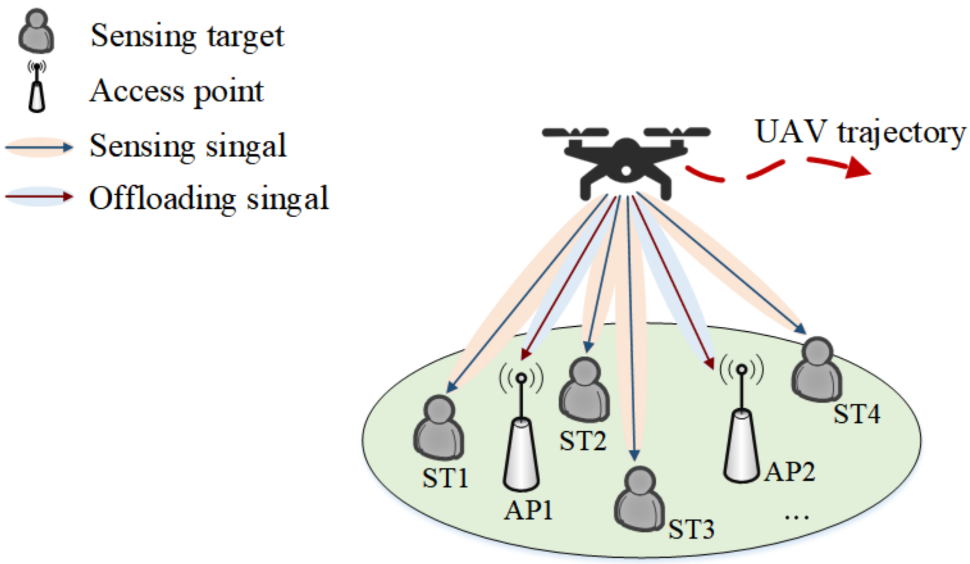

Figure 1. Illustration of the UAV-enabled ISCC network.

The horizontal coordinate of AP n is denoted by $\boldsymbol{q}_n^a = [x_n^a, y_n^a]$. Similarly, the ST k is located at $\boldsymbol{q}_k^s = [x_k^s, y_k^s]$. The horizontal position of the UAV is $\boldsymbol{q}^u(t) = [x^u(t), y^u(t)]$, and its normalized horizontal velocity is $\boldsymbol{v}(t) = [v_x(t), v_y(t)]$ with $\|\boldsymbol{v}(t)\| \leq 1$. The actual velocity is thus $\boldsymbol{V}(t) = V_{\max}\boldsymbol{v}(t)$. The UAV's position evolves as

$$\boldsymbol{q}^u(t+1) = \boldsymbol{q}^u(t) + \delta_T \boldsymbol{V}(t), \quad (1)$$

where δ_T represents the duration of a single slot.

2.1. Communication Model

To characterize the practical air-to-ground channel propagation with multipath effects, we adopt the Rician fading model [17,36]. This model accounts for both the deterministic

LoS component and the non-line-of-sight (NLoS) multipath scattering components. The channel vector between the UAV and AP n at time slot t is modeled as

$$\boldsymbol{h}_n(t) = \frac{\sqrt{\beta_0}}{d_n(t)}\left(\sqrt{\frac{\kappa}{\kappa+1}}\boldsymbol{h}_n^{\text{LoS}}(t) + \sqrt{\frac{1}{\kappa+1}}\boldsymbol{h}_n^{\text{NLoS}}(t)\right), \tag{2}$$

where $d_n(t) = \sqrt{\|\boldsymbol{q}^u(t) - \boldsymbol{q}_n^a\|^2 + H^2}$, β_0 represents the channel power gain at the reference distance, and κ denotes the Rician factor representing the ratio of the power of the LoS component to that of the NLoS component.

The deterministic LoS component $\boldsymbol{h}_n^{\text{LoS}}(t)$ is determined by the array response vector

$$\begin{aligned}\boldsymbol{h}_n^{\text{LoS}}(t) = &\left[1, e^{j\frac{2\pi}{\lambda}d_x\Phi_n(t)}, \ldots, e^{j\frac{2\pi}{\lambda}(M_x-1)d_x\Phi_n(t)}\right]^T \\ &\otimes\left[1, e^{j\frac{2\pi}{\lambda}d_y\Omega_n(t)}, \ldots, e^{j\frac{2\pi}{\lambda}(M_y-1)d_y\Omega_n(t)}\right]^T,\end{aligned} \tag{3}$$

with $\Phi_n(t) = \frac{x^u(t)-x_n^a}{d_n(t)}$ and $\Omega_n(t) = \frac{y^u(t)-y_n^a}{d_n(t)}$. The NLoS component $\boldsymbol{h}_n^{\text{NLoS}}(t) \in \mathbb{C}^{M\times 1}$ accounts for the multipath scattering and is distributed as $\mathcal{CN}(\boldsymbol{0}, \boldsymbol{I}_M)$.

$\boldsymbol{w}_n(t)s_n(t)$ is used to denote the transmit signal from the UAV to AP n, with $\boldsymbol{w}_n(t) \in \mathbb{C}^{M\times 1}$ denoting the beamforming vector and $s_n(t) \sim \mathcal{CN}(0,1)$ being the associated data symbol. The SINR at AP n can be written as

$$\gamma_n(t) = \frac{|\boldsymbol{h}_n^H(t)\boldsymbol{w}_n(t)|^2}{\sum_{i\neq n}|\boldsymbol{h}_n^H(t)\boldsymbol{w}_i(t)|^2 + \sigma^2}, \tag{4}$$

where σ^2 represents the noise power. Hence, the achievable rate of AP n is

$$R_n(t) = B\log_2(1+\gamma_n(t)). \tag{5}$$

2.2. Computing Model

The total amount of computation handled by the UAV comprises locally processed tasks and offloaded tasks:

$$L(t) = \frac{\delta_T}{c}f_o(t) + \delta_T\sum_{n=1}^{N}R_n(t), \tag{6}$$

where $f_o(t)$ is the UAV CPU operation frequency, and c denotes the number of CPU cycles required to process one bit of computation.

2.3. Sensing Model

In this study, a dual-functional shared waveform is employed, where the communication stream is concurrently utilized as the radar probing signal [37]. This unified emission inherently eliminates the transmit-side cross-interference between sensing and communication functionalities. Therefore, the sensing beampattern gain from the UAV to ST k, which characterizes the power of the probing signal steered toward the target, is given by [38]

$$\begin{aligned}\Gamma_k(t) &= \boldsymbol{a}^H(\boldsymbol{q}^u(t), \boldsymbol{q}_k^s)\,\boldsymbol{Q}(t)\,\boldsymbol{a}(\boldsymbol{q}^u(t), \boldsymbol{q}_k^s) \\ &= \boldsymbol{a}^H(\boldsymbol{q}^u(t), \boldsymbol{q}_k^s)\left(\sum_{n=1}^{N}\boldsymbol{w}_n(t)\boldsymbol{w}_n^H(t)\right)\boldsymbol{a}(\boldsymbol{q}^u(t), \boldsymbol{q}_k^s).\end{aligned} \tag{7}$$

Regarding the receive side, the interference caused by target sensing echoes at the ground APs is considered mathematically negligible [39]. This is primarily due to the extremely severe two-way propagation loss experienced by the scattered radar signals, which renders them orders of magnitude weaker compared to the dominant direct air-to-ground communication links. Furthermore, while practical sensing echoes received at the UAV inherently contain environmental clutter and multipath interference, we focus on the dominant LoS return for target detection based on the widely adopted assumption that appropriate clutter suppression techniques are successfully applied.

2.4. Energy Consumption Model

We consider three dominant contributors to the total UAV energy consumption: propulsion energy for sustaining flight, radio transmission energy for communication, and computational energy for onboard processing.

Building upon the propulsion energy model established in Ref. [40], the propulsion power of the UAV is

$$\begin{aligned} P(t) =& P_{\text{blade}}\left(1+\frac{3\|\boldsymbol{V}(t)\|^2}{U_{\text{tip}}^2}\right)+P_{\text{induced}}\left(\sqrt{1+\frac{\|\boldsymbol{V}(t)\|^4}{4v_0^4}}-\frac{\|\boldsymbol{V}(t)\|^2}{2v_0^2}\right)^{\frac{1}{2}} \\ &+\frac{1}{2}d_{\text{fuse}}\rho s_{\text{rotor}}A\|\boldsymbol{V}(t)\|^3. \end{aligned} \tag{8}$$

The relevant parameters of this model, along with their descriptions and numerical values, are summarized in Table 1. Hence, the corresponding flight energy consumption is

$$E^{fl}(t) = \delta_T P(t). \tag{9}$$

The communication energy of the UAV is

$$E^{com}(t) = \delta_T \sum_{n=1}^{N} \|\boldsymbol{w}_n(t)\|^2. \tag{10}$$

The computation energy of the UAV is

$$E^{comp}(t) = \delta_T \gamma_c f_o^3(t). \tag{11}$$

In this equation, the computational power grows cubically with the CPU operating frequency $f_o(t)$, and γ_c represents the capacitance factor [41].

By summing the above components, the total UAV energy consumption is

$$E(t) = E^{fl}(t) + E^{com}(t) + E^{comp}(t). \tag{12}$$

Table 1. UAV-ISCC system parameters.

Parameter	Description	Value
N	AP number	2
K	ST number	3
T	Time slots number	20
B	System bandwidth	1 MHz
σ^2	Noise power	−80 dBm
β_0	Channel gain at 1 m	−30 dB
L	Side length of the square area	400 m

Table 1. *Cont.*

Parameter	Description	Value
H	UAV altitude	100 m
$V_{\max}$	Max UAV speed	60 m/s
δ_T	Time slot duration	1 s
κ	Rician factor	10
c	CPU cycles per bit	10^3 cycles/bit
$F^u_{\max}$	Max UAV CPU frequency	5 GHz
$F^a_{\max}$	Max AP CPU frequency	15 GHz
γ_c	Capacitance factor	10^{-28}
P_w	Max transmit power of UAV	30 dBm
$\Gamma_{\min}$	Min sensing SNR threshold	10^{-5}
E^u	UAV initial energy budget	2×10^4 J
P_{blade}	Blade power	79.86 W
P_{induced}	Induced power	88.63 W
U_{tip}	Tip speed of the rotor blade	120 m/s
v_0	Mean rotor-induced velocity during hovering	4.03 m/s
d_{fuse}	Fuselage drag ratio	0.6
ρ	Air density	1.225 kg/m^3
s_{rotor}	Rotor solidity	0.05
A	Rotor disk area	0.503 m^2
M	Number of UPA antenna elements	4×4
d_x, d_y	Inter-element spacing of UPA	$\lambda/2$

2.5. Problem Formulation

This work focuses on the long-term computing efficiency maximization of the UAV-enabled ISCC system throughout a task horizon. The instantaneous computing efficiency in slot t is quantified by $\frac{L(t)}{E(t)}$. To this end, we optimize the UAV trajectory, beamforming vectors, and CPU operating frequency. The optimization problem is given as

$$P_1: \quad \max_{v(t),\boldsymbol{w}_n(t),f_o(t)} \frac{1}{T}\sum_{t=1}^{T}\frac{L(t)}{E(t)} \tag{13}$$

$$\text{s.t.} \quad \Gamma_k(t) \geq d_k^2(t)\Gamma_{\min}, \quad \forall t,k, \tag{13a}$$

$$0 \leq f_o(t) \leq F^u_{\max}, \quad \forall t, \tag{13b}$$

$$R_n(t)c \leq F^a_{\max}, \quad \forall n,t, \tag{13c}$$

$$\sum_{n=1}^{N}\|\boldsymbol{w}_n(t)\|^2 \leq P_w, \quad \forall t, \tag{13d}$$

$$\sum_{t=1}^{T}E(t) \leq E^u, \tag{13e}$$

$$x^u(t), y^u(t) \in [0, L], \quad \forall t, \tag{13f}$$

$$\|\boldsymbol{q}^u(t+1) - \boldsymbol{q}^u(t)\| \leq V_{\max}\delta_T, \quad \forall t, \tag{13g}$$

where $d_k(t) = \sqrt{\|\boldsymbol{q}^u(t) - \boldsymbol{q}_k^s\|^2 + H^2}$, $F_{\max}^u$ denotes the maximum allowable UAV CPU frequency, and $F_{\max}^a$ is the maximum allowable CPU frequency of APs. Constraint (13a) indicates the sensing gain threshold requirement for each detected target. Constraint (13b) restricts the UAV CPU frequency. Constraint (13c) ensures that the computing tasks offloaded by the UAV do not exceed the processing capacity of the AP. Constraint (13d) imposes an upper bound on UAV transmit power. Constraint (13e) requires that the UAV's total energy consumption remain below its available onboard energy budget. Constraints (13f) and (13g) restrict the UAV's motion within the square region and cap its movement in each time slot. Evidently, the problem P_1 exhibits non-convexity, primarily attributable to the fractional objective function and the non-convex constraint (13c). Moreover, the system dynamics are stochastic and time-varying. These couplings imply that myopic per-slot optimization may yield infeasible or energy-inefficient behaviors over the entire horizon. Therefore, an effective solution should be able to account for long-term consequences and exploit temporal correlations in the system evolution. This motivates the use of a memory-based DRL approach, where an LSTM-enhanced TD3 agent learns a sequential decision policy that jointly adapts trajectory design and resource allocation in a dynamic environment.

3. Proposed Algorithm

3.1. Preliminaries

Reinforcement learning algorithms for continuous control are often built on the actor–critic architecture, where an actor produces actions according to the current state, and a critic computes the corresponding action-value function [42]. The deep deterministic policy gradient (DDPG) is based on the actor–critic framework [43]. Owing to its capability to directly manage continuous action spaces, DDPG has been widely adopted for optimization of UAV networks [44–46]. However, DDPG suffers from Q-value overestimation and an unstable learning process. In practice, these issues often manifest as oscillatory training curves, poor sample efficiency, and performance degradation in noisy or highly dynamic environments such as UAV-enabled ISCC systems.

TD3 extends the DDPG algorithm and is designed to reduce the overestimation bias inherent in actor–critic architectures [47]. Specifically, TD3 maintains two critic networks, Q_1 and Q_2. At the next state s_{t+1}, it uses the minimum of their target values for bootstrapping, which yields a more conservative and stable value estimate. Moreover, TD3 uses target policy smoothing by adding clipped Gaussian noise to the target action before squashing to reduce sensitivity to small perturbations and overfitting to value spikes. It further applies delayed policy updates, updating the actor and target networks only once every few critic updates to improve training stability. These mechanisms jointly improve the stability and robustness of policy learning, and TD3 has been shown to achieve strong performance in UAV-assisted wireless network optimization [48]. However, in highly dynamic UAV-enabled ISCC scenarios, the effective state is often temporally correlated and only partially observable, for example, due to time-varying channels and accumulated energy consumption. Plain TD3 only conditions on the instantaneous state and does not explicitly exploit temporal dependencies. As a result, its decisions can be suboptimal and its convergence slower, particularly in the presence of long-term constraints and non-stationary system dynamics.

LSTM is a kind of recurrent neural network (RNN) [49]. An LSTM unit is built around a memory cell, together with input, forget, and output gates that regulate how information is entered, within, and out of the cell. The memory cell maintains an internal cell state that serves as a long-term information carrier for the sequence. The cell state is updated by combining the previous cell state with a candidate state, where the contributions of each component are regulated by the forget and input gates. The hidden state is then produced by the output gate as a filtered version of the updated cell state. Through these learnable gates, LSTM can adaptively decide which information to retain or discard over time, effectively capturing long-range temporal dependencies and yielding superior performance on sequence modeling tasks compared to standard RNNs [50]. Previous studies have demonstrated that integrating LSTM into DRL methods, such as the TD3 algorithm, can lead to improved performance in solving MDPs and Partially Observable MDPs (POMDPs) [51]. When integrated into DRL, LSTM allows the agent to exploit temporal correlations and partially observable information by summarizing past trajectories into compact hidden states [52], which in turn improves value estimation, stabilizes policy updates, and accelerates convergence in dynamic environments such as UAV-enabled ISCC systems.

3.2. LSTM-TD3-Based Framework

To capture temporal dependencies in UAV-enabled ISCC networks, we propose the LSTM-TD3-based trajectory optimization and resource allocation algorithm named LTTDRA. The architecture of the LTTDRA algorithm is shown in Figure 2. The LSTM–TD3 framework consists of a single pair of actor and two pairs of critic. On the actor side, we maintain an online policy $\pi(s|\phi)$, as well as its target counterpart $\pi'(s|\phi')$. On the critic side, we employ two online Q-networks $Q_j(s,a|\theta_j)$ and their corresponding target networks $Q'_j(s,a|\theta' j)$ for $j = 1,2$. The replay buffer is augmented to store the transition $(s_t, a_t, r_t, s_{t+1}, d_t)$ and the associated past history h^l_t, which is given as

$$h^l_t = \begin{cases} s_{t-l}, a_{t-l}, \ldots, s_{t-1}, a_{t-1}, & \text{if } l \geq 1, \\ s^0, a^0, & \text{otherwise.} \end{cases} \tag{14}$$

where l is the history length, s^0 and a^0 are zero vectors.

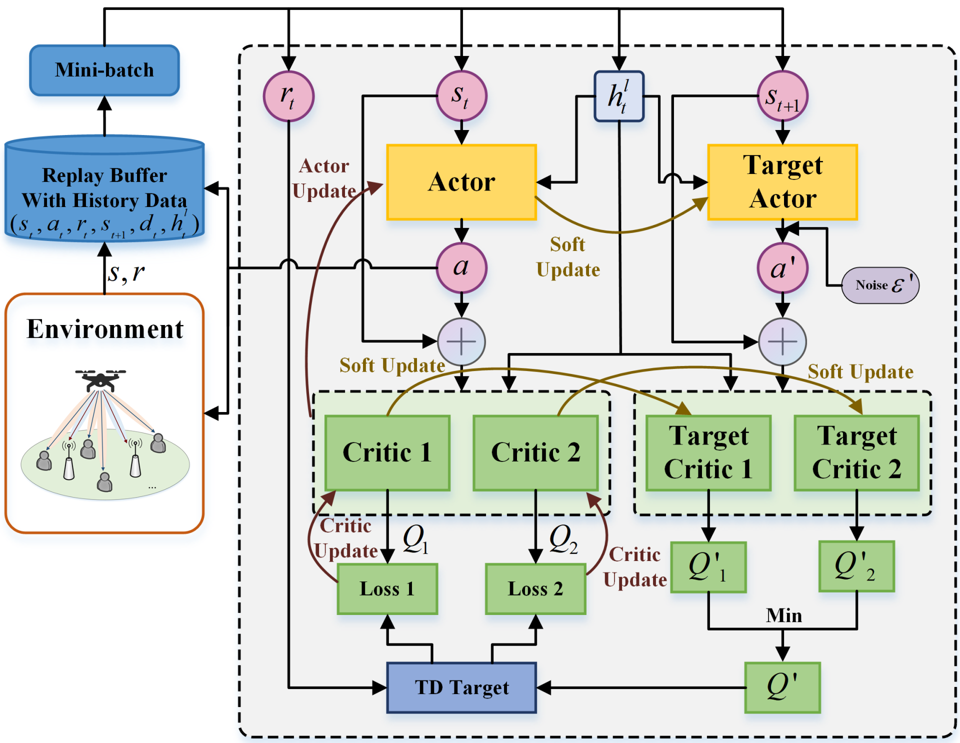

Figure 2. The framework of LTTDRA.

The architectures of the actor and critic are shown in Figure 3. In both networks, an LSTM module is embedded to distill informative temporal features from the history sequence. A memory-based critic Q is utilized to evaluate the Q-value. Specifically, Q is implemented as a composite function with three modules: a memory extraction module Q^{me}, a current feature extraction module Q^{cf}, and a perception integration module Q^{pi}. As described in Ref. [51], it can be expressed as

$$Q(s_t, a_t, h_t^l) = Q^{pi}\left(Q^{me}(h_t^l) \bowtie Q^{cf}(s_t, a_t)\right), \tag{15}$$

where $\bowtie$ denotes the concatenation operator, $Q^{me}(h_t^l)$ represents the memory feature extracted from the historical sequence h_t^l, and $Q^{cf}(s_t, a_t)$ denotes the current feature extracted from the instantaneous state–action pair (s_t, a_t).

Analogously, the actor π consists of a memory extraction module π^{me}, a current feature extraction module π^{cf}, and a perception integration module π^{pi}, which is given as

$$\pi(s_t, h_t^l) = \pi^{pi}\left(\pi^{me}(h_t^l) \bowtie \pi^{cf}(s_t)\right), \tag{16}$$

where $\pi^{me}(h_t^l)$ denotes the memory feature extracted from the history h_t^l, and $\pi^{cf}(s_t)$ denotes the current feature extracted from the present state s_t.

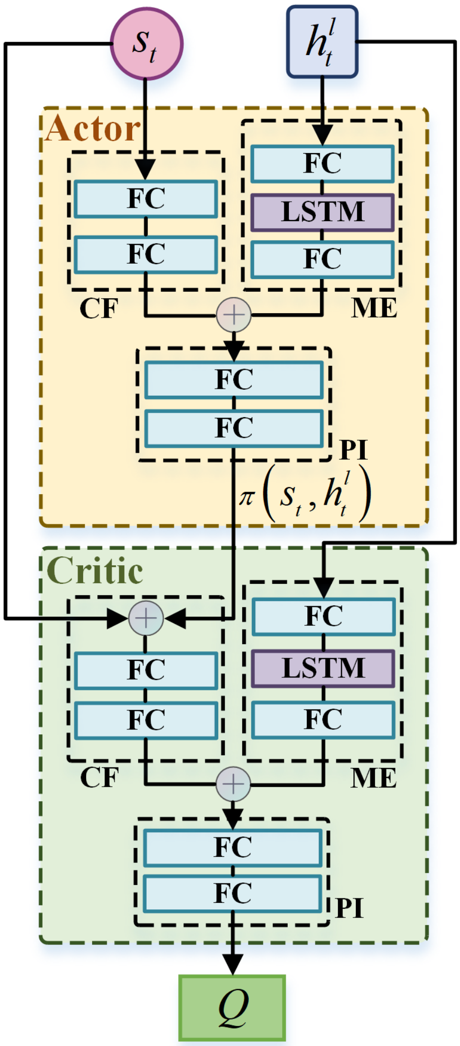

Figure 3. The architectures of the actor and critic networks in LTTDRA.

3.3. DRL Problem Formulation

We cast the proposed problem into an MDP. At every discrete time instant, the UAV agent interacts with the dynamic environment by observing state $s(t)$, executing action $a(t)$, and receiving the immediate reward $r(t)$. Over an episode consisting of T time slots, the UAV agent aims to learn an optimal policy to obtain a maximum accumulated reward, which is designed to reflect the long-term computing efficiency under sensing and energy constraints. The elements of the MDP are defined as follows.

(1) State

The state is represented by a vector that collects the UAV's instantaneous kinematic information, as well as its spatial relations to all APs and STs:

$$s(t) = \left[\boldsymbol{q}^u(t), \boldsymbol{v}(t), \{d_n^{rel}(t)\}_{n=1}^N, \{d_k^{rel}(t)\}_{k=1}^K\right], \tag{17}$$

where $d_n^{rel}(t) = \|\boldsymbol{q}^u(t) - \boldsymbol{q}_n^a\|$ and $d_k^{rel}(t) = \|\boldsymbol{q}^u(t) - \boldsymbol{q}_k^s\|$. Incorporating the UAV's instantaneous position and velocity into the state allows the agent to track its mobility pattern and respect motion constraints, which is essential for planning efficient trajectories over time. The distance terms further provide a compact yet informative description of the large-scale channel and sensing geometry, thereby supporting more effective decision-making in subsequent control steps.

(2) Action

The action of the UAV agent consists of its horizontal velocity, the beamforming vectors, and the CPU operating frequency:

$$a(t) = \left[\boldsymbol{v}(t), \{\boldsymbol{w}_n(t)\}_{n=1}^N, f_o(t)\right], \tag{18}$$

This formulation induces a continuous action space. To facilitate processing by real-valued neural networks, each complex beamforming vector $\boldsymbol{w}_n(t)$ is parameterized by its real and imaginary parts, $\boldsymbol{w}_n^{\mathrm{Re}}(t)$ and $\boldsymbol{w}_n^{\mathrm{Im}}(t)$. Consequently, the beamforming action is represented as the concatenation $\boldsymbol{a}_{\mathrm{beam}}(t) = [\boldsymbol{w}_1^{\mathrm{Re}}(t)^T, \boldsymbol{w}_1^{\mathrm{Im}}(t)^T, \ldots, \boldsymbol{w}_N^{\mathrm{Re}}(t)^T, \boldsymbol{w}_N^{\mathrm{Im}}(t)^T]^T$.

(3) Reward Function

To align the learning objective with the computing efficiency maximization problem in P_1, we design an immediate reward that simultaneously promotes energy-efficient computation and discourages constraint violations through penalty terms. To provide informative learning signals in a model-free setting and enhance convergence, we adopt reward function engineering. The immediate reward at time slot t is given by

$$r(t) = \chi(t)\,\psi_{\mathrm{eff}} \frac{L(t)}{E(t)} - \psi_{\mathrm{b}} G_{\mathrm{b}}(t) - \psi_{\Gamma} G_{\Gamma}(t) - \psi_{\mathrm{c}} G_{\mathrm{c}}(t) - \psi_{\mathrm{e}} G_{\mathrm{e}}(t) - \psi_{\mathrm{eb}} G_{\mathrm{eb}}(t), \tag{19}$$

where $\chi(t) = \mathbf{1}\{(13\mathrm{a}) \text{ holds at } t\}$ is the indicator for the sensing constraint. ψ_{eff}, ψ_{b}, ψ_{Γ}, ψ_{c}, ψ_{e}, ψ_{eb} are constant parameters selected following multiple experimental iterations. In practice, these weighting coefficients balance the exploitation of computing efficiency and the satisfaction of system constraints. The penalty terms of the reward function are defined as

$$G_{\mathrm{b}}(t) = \mathbf{1}\{\, x^u(t) \notin [0, L] \ \vee \ y^u(t) \notin [0, L]\,\}, \tag{20}$$

$$G_{\Gamma}(t) = \sum_k \mathbf{1}\Big\{\Gamma_k(t) < d_k^2(t)\,\Gamma_{\min}\Big\}, \tag{21}$$

$$G_{\mathrm{c}}(t) = \sum_n \max\{R_n(t)c - F_{\max}^a, 0\}, \tag{22}$$

$$G_{\mathrm{e}}(t) = \max\{E(t) - E^{u}_{\mathrm{rem}}(t), 0\}, \tag{23}$$

$$G_{\mathrm{eb}}(t) = \max\{E(t) - E^{u}_{\mathrm{bud}}(t), 0\}. \tag{24}$$

The reward function in Equation (19) consists of six terms. The first term provides a positive reward proportional to the system's computing efficiency, measured by the ratio of effectively processed data $L(t)$ to the total energy consumption $E(t)$. This term directly incentivizes the agent to increase the number of effectively processed bits per joule at each slot, thereby aligning the instantaneous feedback with the desired energy-efficient ISCC operation over the entire horizon. The indicator $\chi(t)$ further enforces sensing reliability by allowing positive efficiency rewards only when the sensing constraint is satisfied, which prevents the agent from pursuing computation gains at the cost of violating the required sensing threshold. The remaining terms impose penalties to discourage constraint violations. $G_{\mathrm{b}}(t)$ penalizes violations of the UAV's operational boundary. $G_{\Gamma}(t)$ penalizes unsatisfied sensing reliability. $G_{\mathrm{c}}(t)$ penalizes computational overload at the APs. The penalty components $G_{\mathrm{b}}(t)$, $G_{\Gamma}(t)$ and $G_{\mathrm{c}}(t)$ provide explicit and interpretable signals for constraint handling, encouraging the learned policy to remain within the service area, maintain reliable sensing performance, and avoid overloading AP computing resources. $G_{\mathrm{e}}(t)$ penalizes violations of the UAV's energy consumption limit. Here, $E^{u}_{\mathrm{rem}}(t) = E^{u} - \sum_{1}^{t} E(\tau)$ represents the remaining onboard energy of the UAV. The last term is a per-slot energy-budget penalty that provides dense guidance and alleviates the sparse punishment caused by the long-term energy constraint. At time slot t, the UAV energy budget is

$$E^{u}_{\mathrm{bud}}(t) = \frac{\max\{E^{u}_{\mathrm{rem}}(t), 0\}}{T - t + 1}. \tag{25}$$

This reward function engineering mechanism transforms an otherwise sparse long-horizon constraint into dense per-step guidance, which improves credit assignment, mitigates premature energy depletion, and empirically accelerates convergence.

The weighting coefficients in the reward function are selected based on the principles of magnitude alignment and hierarchical constraint enforcement. The efficiency coefficient ψ_{eff} acts as a normalizer to scale the reward signal to a manageable range for the neural network, preventing gradient instability. The penalty weights reflect the priority of constraints. Critical "hard constraints" that terminate the episode or cause mission failure, such as the boundary violation penalty ψ_{b} and the energy depletion penalties ψ_{e} and ψ_{eb}, are assigned higher values. This creates a steep gradient barrier, forcing the agent to prioritize safety. Conversely, operational "soft constraints", including the sensing quality weight ψ_{Γ} and the computing capacity weight ψ_{c}, are assigned moderate weights. The specific numerical configurations for these weights are provided in the simulation setup in Section 4.

The above reward design serves as a practical surrogate for the original long-term objective in P_1 while explicitly encoding the feasibility requirements of the UAV-enabled ISCC system. This composite reward function enables the agent to effectively learn feasible and high-performance strategies in complex environments with multiple constraints.

3.4. The Training Procedure

Algorithm 1 summarizes our proposed algorithm. At the beginning of training, two critic networks Q_{θ_1} and Q_{θ_2} and one actor network π_{ϕ} are randomly initialized. The corresponding target networks $Q_{\theta_1'}$, $Q_{\theta_2'}$, and $\pi_{\phi'}$ are then initialized by copying the parameters of their online counterparts. In addition, an experience replay buffer $\mathcal{D}$ is created to store transition tuples together with their associated historical information. The initial state s_1 and the initialized history h_1^l are generated by resetting the environment at the start of every training episode. Subsequently, for each time slot $t = 1, \ldots, T$, the agent selects an action from the current policy augmented with exploration noise:

$$a_t = \pi(s_t, h_t^l) + \varepsilon, \tag{26}$$

where $\varepsilon \sim \mathcal{N}(0, \sigma_a^2)$ is Gaussian noise, and σ_a controls the exploration level. Upon executing a_t in the UAV-enabled ISCC environment, the agent acquires the next state s_{t+1}, the reward r_t, and the terminal flag $d_t \in \{0, 1\}$. In our setting, the terminal flag is triggered as $d_t = 1$ when the episode reaches the final time slot T, or when a terminal constraint such as energy depletion or boundary violation is met. The transition, together with the historical sequence, is then stored in the replay buffer. Subsequently, the history h_{t+1}^l is updated by removing the oldest state–action pair and appending the latest pair (s_t, a_t), as indicated in Algorithm 1, so that a fixed-length sequence of past information is maintained for training. At each gradient update step, a minibatch of N_b transitions, including corresponding histories, is sampled uniformly from $\mathcal{D}$.

Algorithm 1 The proposed LTTDRA

Initialize Q_{θ_1}, Q_{θ_2}, π_ϕ with θ_1, θ_2, ϕ. Set $\theta_1' \leftarrow \theta_1$, $\theta_2' \leftarrow \theta_2$, $\phi' \leftarrow \phi$. Initialize $\mathcal{D}$.
for episode $e = 1 \to E$ **do**
  Reset s_1. Reset h_1^l.
  for $t = 1 \to T$ **do**
    Select the action $a_t = \pi(s_t, h_t^l) + \varepsilon$, where $\varepsilon \sim \mathcal{N}(0, \sigma_a^2)$.
    Execute a_t. Obtain s_{t+1}, r_t, and d_t.
    Store $(s_t, a_t, r_t, s_{t+1}, d_t, h_t^l)$ into $\mathcal{D}$.
    Update history data $h_{t+1}^l \leftarrow (h_t^l - (s_{t-l}, a_{t-l})) \cup (s_t, a_t)$.
    Uniformly sample a minibatch of N_b transitions with their corresponding histories $(s_t, a_t, r_t, s_{t+1}, d_t, h_t^l)$ from $\mathcal{D}$.
    Do target policy smoothing $a' = \pi'(s_{t+1}, h_{t+1}^l) + \varepsilon'$.
    Update critic Q_1 and Q_2 by minimizing (27).
    if t mod d **then**
      Update actor π through maximizing (30).
      Perform soft updates using (31) and (32).
    end if
  end for
end for

For each critic $Q_{j \in \{1,2\}}$, the parameters are updated by minimizing the mean-squared error (MSE) between its predicted value Q_j and the target value $\hat{Q}$:

$$\mathbb{E}\left[(Q_j(s_t, a_t, h_t^l) - \hat{Q})^2\right], \tag{27}$$

where the target value is computed as

$$\hat{Q} = r_t + \gamma(1 - d_t) \min_{j=1,2} Q_j'(s_{t+1}, a', h_{t+1}^l), \tag{28}$$

and the target action is given by

$$a' = \pi'(s_{t+1}, h_{t+1}^l) + \varepsilon', \tag{29}$$

where $\varepsilon' \sim \text{clip}(\mathcal{N}(0, \sigma_{a,t}^2), -c_b, c_b)$ represents a clipped noise with $-c_b$ to c_b bounds.

The actor network is updated by maximizing the expected Q-value provided by the critic Q at the current observation (s_t, h_t^l) and its associated action $\pi(s_t, h_t^l)$:

$$\mathbb{E}\Big[Q_1(s_t, \pi(s_t, h_t^l), h_t^l)\Big]. \tag{30}$$

In the final step, the target actor and critic networks undergo parameter updates based on the soft update mechanism:

$$\phi' \leftarrow \tau\phi + (1-\tau)\phi', \tag{31}$$

$$\theta_j' \leftarrow \tau\theta_j + (1-\tau)\theta_j', \quad j = 1, 2. \tag{32}$$

Through repeated interaction with the environment and stochastic gradient updates, the LSTM–TD3 framework gradually improves the policy. Over time, it converges to a trajectory and resource allocation strategy that achieves high long-term computing efficiency while complying with the imposed system constraints.

3.5. Computational Complexity

The computational cost of the proposed LTTDRA algorithm primarily consists of two components: the training phase and the inference phase. During the training phase, the complexity is dominated by the iterative gradient updates of the actor and critic networks using sampled minibatches from the replay buffer. Compared to the baseline TD3 and DDPG algorithms, LTTDRA incurs a higher training overhead due to the backpropagation through time (BPTT) required for the LSTM modules. Nevertheless, for real-time UAV operation, the complexity is strictly determined by the inference of the actor network. As illustrated in the system architecture, the actor network comprises three modules: the ME module, the CF module, and the PI module. Specifically, the CF and PI modules consist purely of fully connected (FC) layers, while the ME module employs an LSTM layer embedded between FC layers to process the history sequence of length l. The time complexity of the recurrent LSTM layer is strictly bounded by $\mathcal{O}(l \cdot (4H_{lstm}^2 + 4H_{lstm} \cdot D_{in}))$, where H_{lstm} is the number of hidden units and D_{in} is the input dimension of the history sequence. For the FC layers distributed across all three modules, the time complexity is $\mathcal{O}(\sum_{j=1}^{J} n_{j-1} \cdot n_j)$, where J is the total number of FC layers and n_j denotes the number of neurons in layer j.

Compared with standard TD3 and DDPG, which rely solely on standard FC layers, LTTDRA utilizes a multi-branch architecture. Assuming the overall parameter scale of the FC layers is maintained at a comparable level to the baselines, LTTDRA primarily introduces an additional inference complexity of $\mathcal{O}(l \cdot (4H_{lstm}^2 + 4H_{lstm} \cdot D_{in}))$ per decision step. Although this sequence processing slightly increases the inference latency, the network size is moderate. The absolute inference time remains on the order of milliseconds, which is mathematically negligible compared to the macroscopic task time slot duration of $\delta_T = 1$ s. Thus, LTTDRA perfectly meets the stringent requirements for real-time UAV deployment.

4. Simulation Results and Discussion

In this section, we present and analyze simulation results to demonstrate the performance of LTTDRA. The key system parameters of the UAV-enabled ISCC network are listed in Table 1. The hyperparameters for the DRL-based algorithm and the constant coefficients in the reward function are summarized in Table 2 and Table 3, respectively. We adopt the following state-of-the-art DRL-based schemes as benchmark methods: (1) the TD3-based algorithm used in Ref. [48]; (2) the DDPG-based algorithm used in Ref. [46]. The following two baseline schemes are also selected for comparison: (1) Random trajectory algorithm

(RTA): The UAV follows a randomly generated trajectory, while the allocation of remaining resources is optimized using a greedy algorithm; (2) Fixed trajectory algorithm (FTA): The UAV follows a predetermined square trajectory at a constant speed throughout the flight period with a random action selection policy.

Table 2. Hyperparameters of DRL.

Parameter	Meaning	Value
$\mathcal{D}$	Replay buffer size	10^5
N_b	Batch size	256
lr_a	Learning rate for actor networks	10^{-3}
lr_c	Learning rate for critic networks	10^{-3}
σ_a	Actor exploration noise	0.1
$\sigma_{a,t}$	Target actor noise	0.2
c_b	Noise clip boundary	0.5
d	Policy update delay	2
γ	Reward discount factor	0.95
Activation	Activation function	ReLU
Optimizer	Optimization algorithm	Adam
l	History length	3

Table 3. Constant parameters.

Parameter	Value
ψ_{eff}	10^{-3}
ψ_{b}	10^2
ψ_{Γ}	10
ψ_{c}	10
ψ_{e}	10^2
ψ_{eb}	10^2

In Figure 4, we compare the mean accumulated rewards achieved by the LTTDRA algorithm, TD3, and DDPG versus the number of episodes during the training. All three algorithms eventually converge, but LTTDRA consistently achieves the best performance. In the early stage, the mean rewards of LTTDRA rise much more rapidly than those of the other two algorithms, indicating faster learning and more efficient exploration. This gain mainly comes from the memory-based structure, which leverages historical state–action information to capture temporal correlations in system dynamics, enabling the agent to infer more informative latent states and make less myopic decisions. In contrast, the TD3-based algorithm only relies on the current state and therefore reacts more locally to instantaneous observations, which leads to slower convergence and a lower reward plateau. The DDPG-based algorithm performs worse because it lacks the twin–critic structure, target policy smoothing, and delayed policy updates of the TD3-based algorithm, making it more prone to value overestimation and training instability.

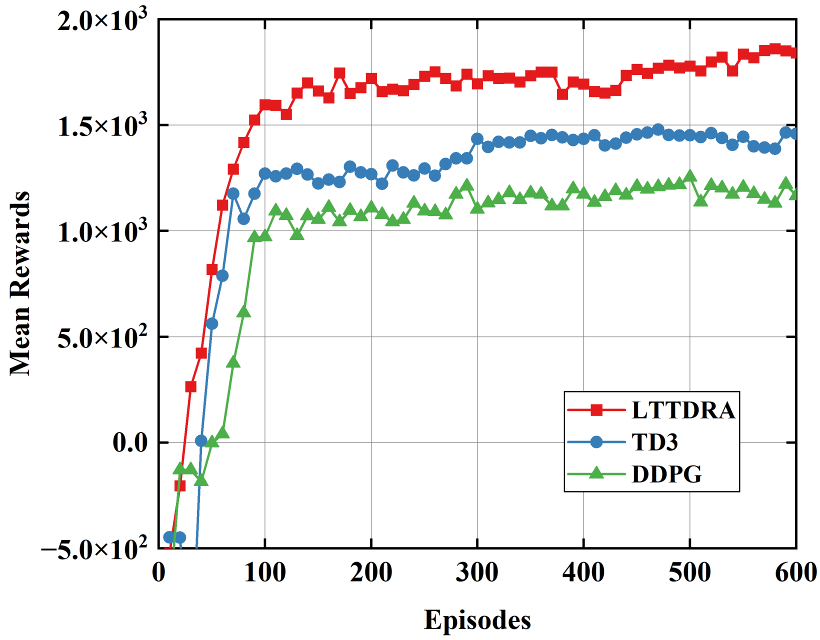

Figure 4. Training curves of the LTTDRA algorithm, TD3, and DDPG.

In Figure 5, we compare the average computing efficiency achieved by the LTTDRA algorithm, TD3, and DDPG versus the number of episodes during the training. All three algorithms achieve a rapid rise in computing efficiency during the initial training phase. Thereafter, the LTTDRA algorithm stabilizes at the highest efficiency level, around 7.87×10^4 Bits/J, while the TD3-based algorithm converges to a lower plateau near 6.86×10^4 Bits/J and the DDPG-based algorithm settles at about 5.88×10^4 Bits/J. Compared with TD3 and DDPG, the LTTDRA algorithm achieves improvements of 14.73% and 33.82%, respectively, in terms of computing efficiency. The higher steady-state efficiency and relatively smaller fluctuations of LTTDRA indicate that exploiting temporal feature extraction via LSTM leads to more energy-efficient and stable computation policies than conventional TD3 and DDPG.

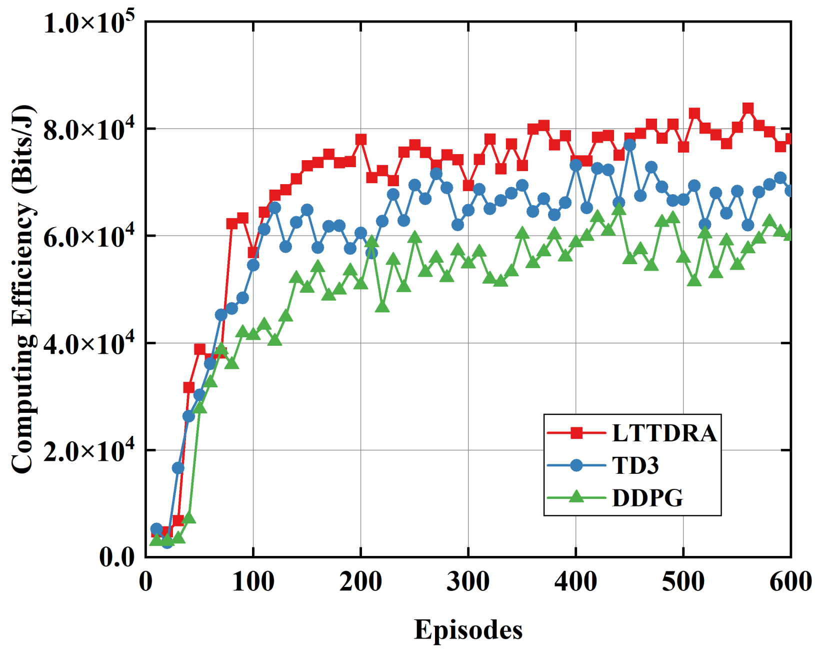

Figure 5. Computing efficiency according to training episodes of the LTTDRA algorithm, TD3, and DDPG.

Notably, the performance comparison between the proposed LTTDRA and the standard TD3 algorithm serves as an ablation study to isolate the contribution of the LSTM architecture. Since standard TD3 lacks a recurrent memory mechanism and processes only instantaneous observations, the superior convergence performance and computing efficiency of LTTDRA, as demonstrated in Figures 4 and 5, directly validate the effectiveness of the LSTM-based temporal feature extraction in dynamic UAV-ISCC environments.

Figure 6 conducts a sensitivity analysis on the history length $l \in \{1,2,3,4,5\}$ for the learning performance of the proposed LTTDRA algorithm. Fundamentally, incorporating historical information sequences empowers the LSTM module to continuously track the temporal dynamics. As observed, $l = 3$ achieves the optimal balance between temporal modeling capability and training efficiency, yielding the fastest convergence and the highest converged reward. When the history length further increases to $l = 5$, the performance slightly degrades compared to $l = 3$, although it still maintains a relatively high reward and ranks second among all tested configurations. This degradation stems from the redundant computational overhead at this simulation step size, hindering optimal policy search. Conversely, a strictly limited history of $l = 1$ and $l = 2$ prevents the agent from capturing essential temporal dynamics, severely degrading performance. And the complex gating mechanisms merely introduce redundant computational overhead. Notably, LTTDRA with $l = 1$ underperforms the memoryless TD3 baseline, as the redundant computational overhead exacerbates training difficulty compared to the standard Multi-Layer Perceptron (MLP) in TD3.

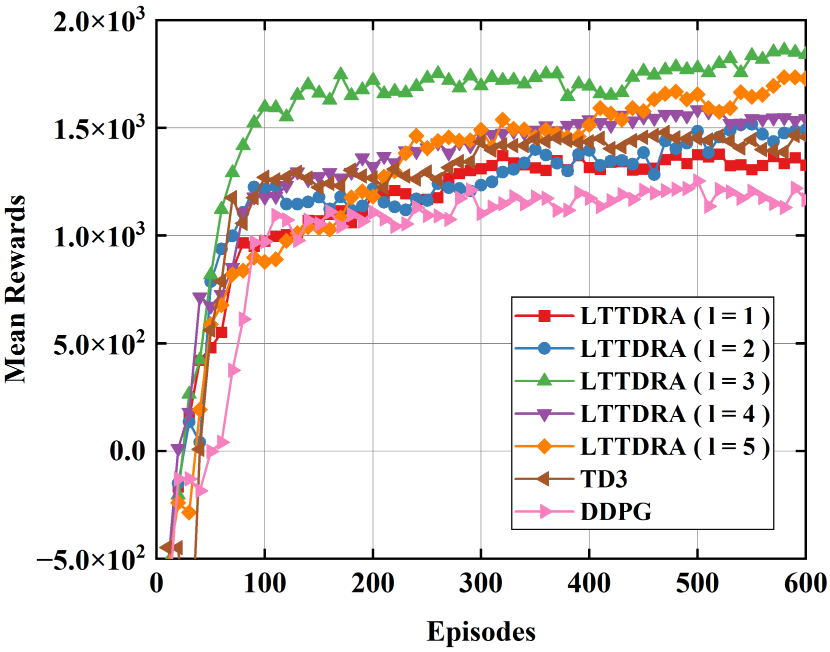

Figure 6. Training curves of the LTTDRA algorithm with different history lengths l.

Figure 7 shows the comparisons of average computing efficiency of the LTTDRA algorithm, TD3, DDPG, RTA, and FTA in various scenarios. Figure 7a depicts the computing efficiency with respect to the number of time slots achieved by LTTDRA and the benchmarks. LTTDRA consistently achieves the highest efficiency, with a clear margin over the baselines. Under each task time, the average computing efficiency achieved by LTTDRA is, respectively, 16.46%, 38.33%, 109.40% and 235.65% higher than that of TD3, DDPG, RTA, and FTA. In Figure 7b, computing efficiency decreases as the UAV altitude increases, since larger link distances degrade both sensing and communication performance, but the LTTDRA algorithm remains superior across all heights. At varied UAV flight altitudes, LTTDRA achieves average computing efficiency improvements of 15.96%, 39.26%, 112.78%, and 253.94% over TD3, DDPG, RTA, and FTA, respectively. Figure 7c indicates that computing efficiency benefits from more APs, as additional APs significantly improve computing efficiency due to enhanced offloading opportunities. In Figure 7c, LTTDRA achieves average computing efficiency improvements of 15.55%, 34.34%, 96.17%, and 226.09% over TD3, DDPG, RTA, and FTA, respectively. Figure 7d indicates that computing efficiency benefits from more STs, as additional sensing tasks help utilize UAV resources more effectively. In Figure 7d, LTTDRA achieves average computing efficiency improvements of 14.29%, 36.15%, 101.73%, and 217.88% over TD3, DDPG, RTA, and FTA, respectively. In all scenarios, LTTDRA consistently outperforms various benchmark approaches, demonstrating outstanding performance. These results confirm the robustness and effectiveness of LTTDRA in improving computing efficiency.

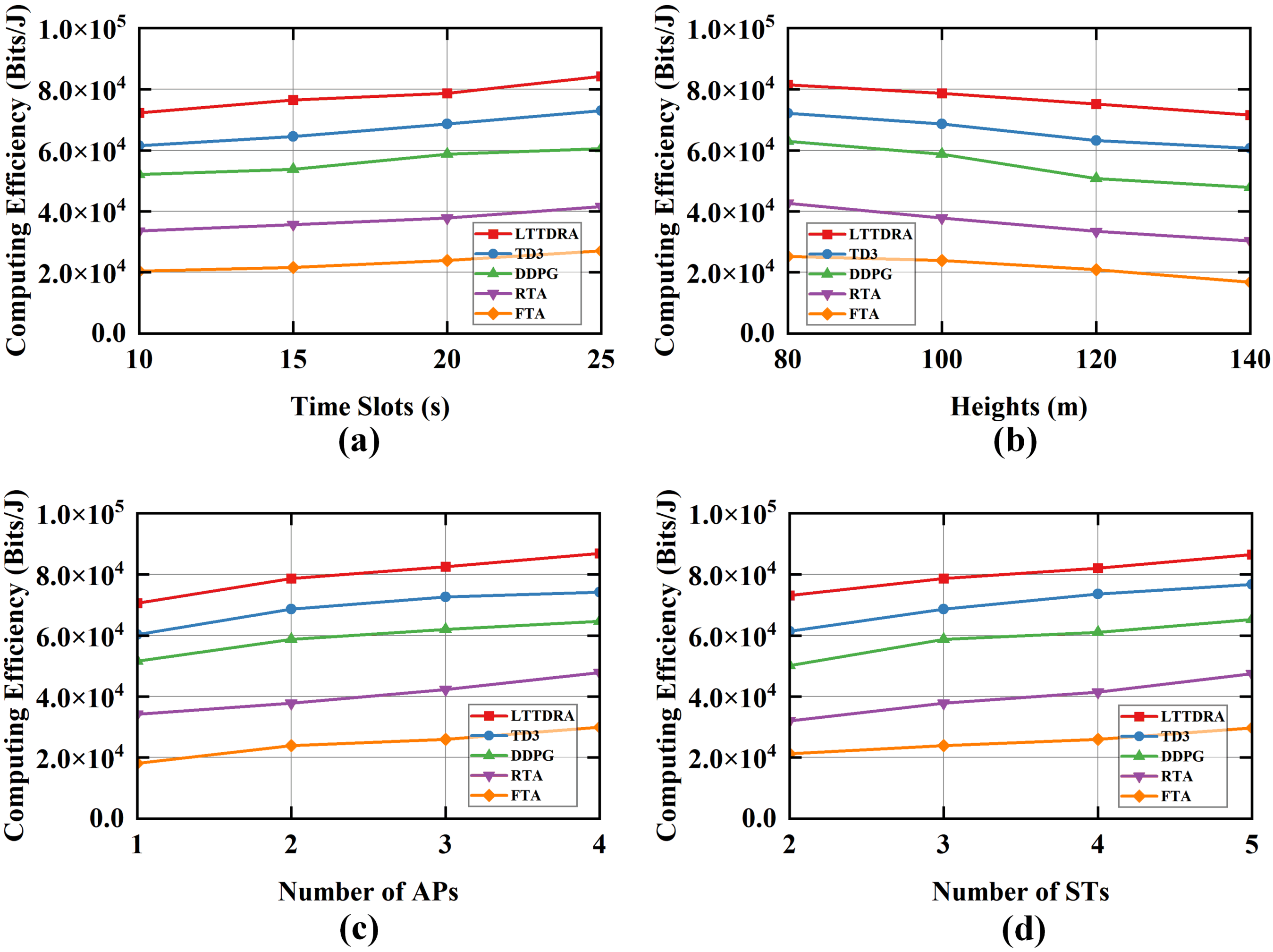

Figure 7. Comparisons of the performance of the LTTDRA algorithm, TD3, DDPG, RTA, and FTA. in various scenarios. (**a**) Different task times. (**b**) Different UAV altitudes. (**c**) Different AP numbers ($K = 3$). (**d**) Different ST numbers ($N = 2$).

Figure 8 depicts the UAV trajectories obtained by the proposed LTTDRA and the benchmark algorithms. The LTTDRA algorithm learns a remarkably smoother and more purposeful path. Under the rotary-wing propulsion model, this smooth trajectory effectively mitigates severe speed fluctuations and minimizes redundant propulsion power consumption, yielding the most favorable tradeoff between ISCC performance and energy expenditure. Consequently, LTTDRA achieves the highest computing efficiency. By contrast, the DDPG and TD3 algorithms fail to achieve such an optimal tradeoff due to their inherent algorithmic constraints. DDPG suffers from severe training variance, producing a highly torturous path with frequent, inefficient steering that consumes excessive propulsion energy. Conversely, while TD3 stabilizes the learning process to avoid extreme detours, it lacks the temporal memory required for long-term spatial planning and resource allocation for cooperative sensing and computing. The RTA exhibits chaotic and undirected flight behaviors, wasting substantial propulsion energy. The FTA follows a rigid, pre-defined trajectory that completely ignores the distribution of STs and APs, inherently failing to exploit spatial proximity.

Overall, the above results consistently verify the effectiveness of the proposed LTTDRA algorithm for computing efficiency maximization in UAV-enabled ISCC systems. Compared with the TD3-based and DDPG-based baselines, LTTDRA exhibits faster convergence and achieves higher accumulated rewards, indicating improved learning efficiency and policy stability enabled by memory-based temporal feature extraction. In terms of computing efficiency, LTTDRA provides significant gains over all benchmarks. Moreover, extensive evaluations under different task durations, UAV altitudes, and network scales demonstrate that LTTDRA maintains consistent advantages, highlighting its robustness to scenario variations. Finally, the trajectory comparison reveals that LTTDRA learns smoother flight paths that simultaneously approach beneficial sensing and communication locations while reducing propulsion energy. These findings confirm that jointly exploiting temporal correlations and coordinated trajectory and resource control is crucial for UAV-ISCC operation.

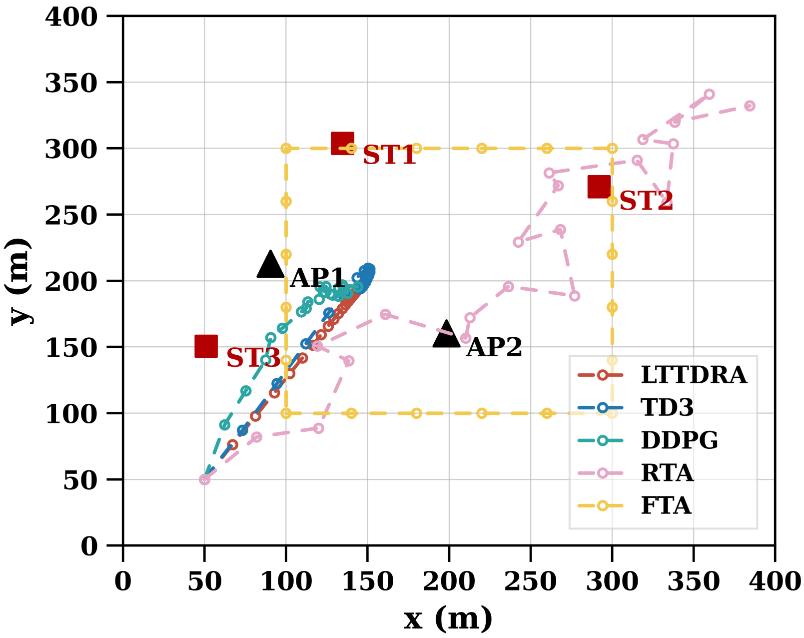

Figure 8. The UAV trajectory comparison of the LTTDRA algorithm, TD3, DDPG, RTA, and FTA.

While the proposed LTTDRA algorithm demonstrates robustness across varying numbers of APs and STs, scaling this framework to ultra-dense networks or multi-UAV scenarios introduces additional complexities. In denser networks, the dimensions of the state and action spaces expand with the number of nodes. To prevent the curse of dimensionality and maintain learning efficiency, future implementations could integrate attention mechanisms to dynamically filter and prioritize the most critical environmental features [30]. Furthermore, transitioning to a multi-UAV architecture transforms the system into a cooperative multi-agent environment. This transition requires managing mutual interference and addressing non-stationary environment dynamics. Such challenges can be effectively resolved by extending the current framework into a multi-agent reinforcement learning (MARL) paradigm, utilizing a centralized training and decentralized execution (CTDE) architecture to enable scalable, interference-aware, and collision-free operations [53].

5. Conclusions

In this paper, we propose a UAV-enabled ISCC network and address the optimization of computing efficiency within this framework. We establish a framework that simultaneously optimizes the UAV trajectory, beamforming vector, and CPU operating frequency, aiming to maximize the computing efficiency subject to sensing reliability and system operation constraints. Fundamentally, a trade-off exists between sensing reliability and computing efficiency in UAV-enabled ISCC networks. Guaranteeing high sensing reliability typically requires the UAV to maneuver closer to the STs and allocate sufficient sensing power. However, maximizing computing efficiency demands minimizing propulsion energy and establishing strong communication links with ground APs to accelerate task offloading. Consequently, prioritizing sensing reliability inherently restricts the UAV's trajectory and consumes power that could otherwise be dedicated to data transmission or local computation, thereby constraining the overall computing efficiency. Balancing this intricate trade-off requires a meticulous joint design of the UAV's trajectory and multi-dimensional resource allocation.

To realize this joint design, the resulting nonconvex optimization problem is cast as an MDP. To address the formulated MDP, we adopt a memory-based DRL framework and propose an LSTM-TD3-based trajectory design and resource allocation algorithm named LTTDRA. LSTM units are embedded into the actor and the critic to extract temporal features from historical state–action sequences. This design allows the agent to capture temporal dependencies in the system dynamics more effectively. The state space and action space are carefully designed to ensure effective representation of environmental dynamics and agent behavior. In addition, the reward function is meticulously designed to alleviate sparse-penalty effects and learn high-performance strategies in complex environments with multiple constraints. Simulation results verify the effectiveness of LTTDRA and show that it significantly outperforms all baseline schemes.

Despite the promising performance of the LTTDRA algorithm, several limitations should be acknowledged for practical deployment. A primary limitation is that our model assumes a steady-state atmospheric environment. In real-world scenarios, stochastic wind fields can significantly alter the UAV's propulsion power consumption as the motors must compensate for wind resistance to maintain the planned trajectory. Furthermore, adverse weather conditions such as rain, fog, or snow can lead to increased atmospheric attenuation and clutter interference. These factors not only degrade the SINR for communication but also reduce the sensing beampattern gain, potentially challenging the reliability of ISCC services. Future research on the UAV-enabled ISCC framework will focus on three main directions. First, the single-UAV model will be extended to a multi-UAV cooperative network. Second, the framework will be adapted for complex 3D flight environments requiring obstacle avoidance. Finally, real-world experimental validations will be conducted as a critical step toward practical deployment. These experiments will bridge the gap between theoretical simulations and practical implementation by addressing real-world constraints, such as onboard processing jitter, flight dynamics uncertainties, and unpredictable weather conditions.

Author Contributions: Conceptualization, H.Q. and M.W.; methodology, H.Q. and M.W.; software, H.Q.; validation, H.Q. and M.W.; formal analysis, H.Q. and M.W.; investigation, H.Q. and M.W.; resources, H.Q. and M.W.; data curation, H.Q. and M.W.; writing—original draft preparation, H.Q. and M.W.; writing—review and editing, H.Q. and M.W.; visualization, H.Q. and M.W.; supervision,

H.Q. and M.W.; project administration, H.Q. and M.W.; funding acquisition, M.W. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Ministry of Education, China-111 project (NO.B17007) and the Director Funds of Beijing Key Laboratory of Network System Architecture and Convergence (NO.2025BKL-NSAC-ZJ-01).

Data Availability Statement: The raw data supporting the conclusions of this article will be made available by the corresponding author on request.

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

Abbreviations

The following abbreviations are used in this manuscript:

AO	Alternating optimization
AP	Access point
BPTT	Backpropagation through time
BS	Base station
CF	Current feature
CTDE	Centralized training and decentralized execution
DDPG	Deep Deterministic Policy Gradient
DRL	Deep reinforcement learning
FC	Fully connected
FTA	Fixed trajectory algorithm
GBS	Ground base station
IoT	Internet of Things
ISAC	Integrated sensing and communication
ISCC	Integrated sensing, computing, and communication
LoS	Line-of-sight
LSTM	Long short-term memory
LTTDRA	LSTM-TD3-based trajectory design and resource allocation
MARL	Multi-agent reinforcement learning
MDP	Markov decision process
ME	Memory extraction
MEC	Mobile edge computing
MSE	Mean-squared error
MSTD	Multi-stage trajectory design
NLoS	Non-line-of-sight
PI	Perception integration
POMDP	Partially observable Markov decision process
RNN	Recurrent neural network
RSMA	Rate-splitting multiple access
RTA	Random trajectory algorithm
SAC	Soft actor–critic
SINR	Signal-to-interference-plus-noise ratio
ST	Sensing target
TD3	Twin delayed deep deterministic policy gradient
TDM	Time-division multiplexing
UAV	Unmanned aerial vehicle
UE	User equipment
UPA	Uniform planar array

References

1. Dong, F.; Liu, F.; Lu, S.; Xiong, Y.; Zhang, Q.; Feng, Z.; Gao, F. Communication-Assisted Sensing in 6G Networks. *IEEE J. Sel. Areas Commun.* **2025**, *43*, 1371–1386. [CrossRef]
2. Feng, Z.; Wei, Z.; Chen, X.; Yang, H.; Zhang, Q.; Zhang, P. Joint Communication, Sensing, and Computation Enabled 6G Intelligent Machine System. *IEEE Netw.* **2021**, *35*, 34–42. [CrossRef]
3. Liu, Y.; Zhang, S.; Mu, X.; Ding, Z.; Schober, R.; Al-Dhahir, N.; Hossain, E.; Shen, X. Evolution of NOMA Toward Next Generation Multiple Access (NGMA) for 6G. *IEEE J. Sel. Areas Commun.* **2022**, *40*, 1037–1071. [CrossRef]
4. Liu, F.; Cui, Y.; Masouros, C.; Xu, J.; Han, T.X.; Eldar, Y.C.; Buzzi, S. Integrated Sensing and Communications: Toward Dual-Functional Wireless Networks for 6G and Beyond. *IEEE J. Sel. Areas Commun.* **2022**, *40*, 1728–1767. [CrossRef]
5. Wei, Z.; Liu, F.; Masouros, C.; Su, N.; Petropulu, A.P. Toward Multi-Functional 6G Wireless Networks: Integrating Sensing, Communication, and Security. *IEEE Commun. Mag.* **2022**, *60*, 65–71. [CrossRef]
6. Zhang, Z.; Xiao, Y.; Ma, Z.; Xiao, M.; Ding, Z.; Lei, X.; Karagiannidis, G.K.; Fan, P. 6G Wireless Networks: Vision, Requirements, Architecture, and Key Technologies. *IEEE Veh. Technol. Mag.* **2019**, *14*, 28–41. [CrossRef]
7. Wen, D.; Li, X.; Zhou, Y.; Shi, Y.; Wu, S.; Jiang, C. Integrated Sensing-Communication-Computation for Edge Artificial Intelligence. *IEEE Internet Things Mag.* **2024**, *7*, 14–20. [CrossRef]
8. Wen, D.; Zhou, Y.; Li, X.; Shi, Y.; Huang, K.; Letaief, K.B. A Survey on Integrated Sensing, Communication, and Computation. *IEEE Commun. Surv. Tutor.* **2025**, *27*, 3058–3098. [CrossRef]
9. Zhao, Y.; Wu, Q.; Chen, W.; Zeng, Y.; Liu, R.; Mei, W.; Hou, F.; Ma, S. Multi-Functional Beamforming Design for Integrated Sensing, Communication, and Computation. *IEEE Trans. Commun.* **2025**, *73*, 6322–6336. [CrossRef]
10. Luo, H.; Wang, Y.; Luo, D.; Zhao, J.; Wu, H.; Ma, S.; Gao, F. Integrated Sensing and Communications in Clutter Environment. *IEEE Trans. Wirel. Commun.* **2024**, *23*, 10941–10956. [CrossRef]
11. Jiang, Y.; Li, X.; Zhu, G.; Li, H.; Deng, J.; Han, K.; Shen, C.; Shi, Q.; Zhang, R. Integrated Sensing and Communication for Low Altitude Economy: Opportunities and Challenges. *IEEE Commun. Mag.* **2025**, *63*, 72–78. [CrossRef]
12. Feng, Y.; Zhao, C.; Luo, H.; Gao, F.; Liu, F.; Jin, S. Networked ISAC based UAV Tracking and Handover Towards Low-Altitude Economy. *IEEE Trans. Wirel. Commun.* **2025**, *24*, 7670–7685. [CrossRef]
13. Qi, H.; Wu, M.; Zhang, Z.; Zhao, M. Trajectory Design for Multi-UAV-Enabled Wireless Powered Communication Networks: A Multi-Agent DRL Approach. In Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC), Dubai, United Arab Emirates, 21–23 April 2024; pp. 1–6. [CrossRef]
14. Tang, J.; Yu, Y.; Pan, C.; Ren, H.; Wang, D.; Wang, J.; You, X. Cooperative ISAC-Empowered Low-Altitude Economy. *IEEE Trans. Wirel. Commun.* **2025**, *24*, 3837–3853. [CrossRef]
15. Wu, Q.; Xu, J.; Zeng, Y.; Ng, D.W.K.; Al-Dhahir, N.; Schober, R.; Swindlehurst, A.L. A Comprehensive Overview on 5G-and-Beyond Networks With UAVs: From Communications to Sensing and Intelligence. *IEEE J. Sel. Areas Commun.* **2021**, *39*, 2912–2945. [CrossRef]
16. Wei, Z.; Zhu, M.; Zhang, N.; Wang, L.; Zou, Y.; Meng, Z.; Wu, H.; Feng, Z. UAV-Assisted Data Collection for Internet of Things: A Survey. *IEEE Internet Things J.* **2022**, *9*, 15460–15483. [CrossRef]
17. Li, B.; Liu, W.; Xie, W.; Zhang, N.; Zhang, Y. Adaptive Digital Twin for UAV-Assisted Integrated Sensing, Communication, and Computation Networks. *IEEE Trans. Green Commun. Netw.* **2023**, *7*, 1996–2009. [CrossRef]
18. Liu, Y.; Xiong, K.; Ni, Q.; Fan, P.; Letaief, K.B. UAV-Assisted Wireless Powered Cooperative Mobile Edge Computing: Joint Offloading, CPU Control, and Trajectory Optimization. *IEEE Internet Things J.* **2020**, *7*, 2777–2790. [CrossRef]
19. Liu, Y.; Li, X.; He, B.; Gu, M.; Huangfu, W. UAV-enabled diverse data collection via integrated sensing and communication functions based on deep reinforcement learning. *Drones* **2024**, *8*, 647. [CrossRef]
20. Hu, H.; Xiong, K.; Qu, G.; Ni, Q.; Fan, P.; Letaief, K.B. AoI-Minimal Trajectory Planning and Data Collection in UAV-Assisted Wireless Powered IoT Networks. *IEEE Internet Things J.* **2021**, *8*, 1211–1223. [CrossRef]
21. Worku, Y.M.; Christodoulou, C.; Devetsikiotis, M. UAV-Mounted Base Station Coverage and Trajectory Optimization Using LSTM-A2C with Attention. *Drones* **2025**, *9*, 787. [CrossRef]
22. Zhao, J.; Gao, F.; Jia, W.; Yuan, W.; Jin, W. Integrated Sensing and Communications for UAV Communications With Jittering Effect. *IEEE Wirel. Commun. Lett.* **2023**, *12*, 758–762. [CrossRef]
23. Liu, Y.; Liu, S.; Liu, X.; Liu, Z.; Durrani, T.S. Sensing Fairness-Based Energy Efficiency Optimization for UAV Enabled Integrated Sensing and Communication. *IEEE Wirel. Commun. Lett.* **2023**, *12*, 1702–1706. [CrossRef]
24. Feng, J.; Liu, X.; Liu, Z.; Liu, Y. Trajectory Design and Resource Allocation for RSMA-UAV Enabled Integrated Sensing and Communication Networks. *IEEE Trans. Cogn. Commun. Netw.* **2026**, *12*, 1241–1254. [CrossRef]
25. Jing, X.; Liu, F.; Masouros, C.; Zeng, Y. ISAC From the Sky: UAV Trajectory Design for Joint Communication and Target Localization. *IEEE Trans. Wirel. Commun.* **2024**, *23*, 12857–12872. [CrossRef]
26. Wang, J.; Zhang, X.; Wang, Y.; Yao, X.; Wei, Z.; Sun, F.; Feng, Z. Transmit-Receive Beamforming for ISAC-Enabled Multi-UAVs System. *IEEE Trans. Green Commun. Netw.* **2026**, *10*, 652–666. [CrossRef]

27. Xu, Y.; Zhang, T.; Liu, Y.; Yang, D. UAV-Enabled Integrated Sensing, Computing, and Communication: A Fundamental Trade-Off. *IEEE Wirel. Commun. Lett.* **2023**, *12*, 843–847. [CrossRef]
28. Zhou, Y.; Liu, X.; Zhai, X.; Zhu, Q.; Durrani, T.S. UAV-Enabled Integrated Sensing, Computing, and Communication for Internet of Things: Joint Resource Allocation and Trajectory Design. *IEEE Internet Things J.* **2024**, *11*, 12717–12727. [CrossRef]
29. Liu, Z.; Liu, X.; Yang, W.; Zhang, X. Joint Sensing and Age of Information Optimization for Energy Constrained UAV-Assisted Integrated Sensing, Calculation, and Communication. *IEEE Trans. Wirel. Commun.* **2025**, *24*, 4440–4453. [CrossRef]
30. Zhu, Y.; Yang, B.; Liu, M.; Li, Z. UAV Trajectory Optimization for Large-Scale and Low-Power Data Collection: An Attention-Reinforced Learning Scheme. *IEEE Trans. Wirel. Commun.* **2024**, *23*, 3009–3024. [CrossRef]
31. Chen, M.; Challita, U.; Saad, W.; Yin, C.; Debbah, M. Artificial Neural Networks-Based Machine Learning for Wireless Networks: A Tutorial. *IEEE Commun. Surv. Tutor.* **2019**, *21*, 3039–3071. [CrossRef]
32. Zhang, Z.; Xu, C.; Li, Z.; Zhao, X.; Wu, R. Deep Reinforcement Learning for Aerial Data Collection in Hybrid-Powered NOMA-IoT Networks. *IEEE Internet Things J.* **2023**, *10*, 1761–1774. [CrossRef]
33. Wu, D.; Wang, L.; Liang, M.; Kang, Y.; Jiao, Q.; Cheng, Y.; Li, J. UAV-Assisted Real-Time Video Transmission for Vehicles: A Soft Actor–Critic DRL Approach. *IEEE Internet Things J.* **2024**, *11*, 14710–14726. [CrossRef]
34. Yan, K.; Xiang, L.; Zheng, K.; Hu, J.; Yang, K.; Liu, J. UAV-Assisted Integrated Active and Passive Sensing with Communication: A Hybrid-SAC Approach. *IEEE Trans. Veh. Technol.* 2025, *early access*. [CrossRef]
35. Song, F.; Deng, M.; Xing, H.; Liu, Y.; Ye, F.; Xiao, Z. Energy-Efficient Trajectory Optimization With Wireless Charging in UAV-Assisted MEC Based on Multi-Objective Reinforcement Learning. *IEEE Trans. Mob. Comput.* **2024**, *23*, 10867–10884. [CrossRef]
36. Khuwaja, A.A.; Chen, Y.; Zhao, N.; Alouini, M.S.; Dobbins, P. A Survey of Channel Modeling for UAV Communications. *IEEE Commun. Surv. Tutor.* **2018**, *20*, 2804–2821. [CrossRef]
37. Liu, F.; Masouros, C.; Petropulu, A.P.; Griffiths, H.; Hanzo, L. Joint Radar and Communication Design: Applications, State-of-the-Art, and the Road Ahead. *IEEE Trans. Commun.* **2020**, *68*, 3834–3862. [CrossRef]
38. Stoica, P.; Li, J.; Xie, Y. On Probing Signal Design For MIMO Radar. *IEEE Trans. Signal Process.* **2007**, *55*, 4151–4161. [CrossRef]
39. Zhang, J.A.; Liu, F.; Masouros, C.; Heath, R.W.; Feng, Z.; Zheng, L.; Petropulu, A. An Overview of Signal Processing Techniques for Joint Communication and Radar Sensing. *IEEE J. Sel. Top. Signal Process.* **2021**, *15*, 1295–1315. [CrossRef]
40. Zeng, Y.; Xu, J.; Zhang, R. Energy Minimization for Wireless Communication With Rotary-Wing UAV. *IEEE Trans. Wirel. Commun.* **2019**, *18*, 2329–2345. [CrossRef]
41. Chen, Y.; Kang, H.; Li, J.; Sun, G.; Wang, B.; Wang, J.; Liang, C.; Liang, S.; Niyato, D. Joint Resource Management for Energy-Efficient UAV-Assisted SWIPT-MEC: A Deep Reinforcement Learning Approach. *IEEE Internet Things J.* **2025**, *12*, 31448–31465. [CrossRef]
42. Haklidir, M.; Temeltaş, H. Guided Soft Actor Critic: A Guided Deep Reinforcement Learning Approach for Partially Observable Markov Decision Processes. *IEEE Access* **2021**, *9*, 159672–159683. [CrossRef]
43. Lillicrap, T.; Hunt, J.; Pritzel, A.; Heess, N.; Erez, T.; Tassa, Y.; Silver, D.; Wierstra, D. Continuous control with deep reinforcement learning. In Proceedings of the 4th International Conference on Learning Representations (ICLR), San Juan, Puerto Rico, 2–4 May 2016.
44. Yu, Y.; Tang, J.; Huang, J.; Zhang, X.; So, D.K.C.; Wong, K.K. Multi-Objective Optimization for UAV-Assisted Wireless Powered IoT Networks Based on Extended DDPG Algorithm. *IEEE Trans. Commun.* **2021**, *69*, 6361–6374. [CrossRef]
45. Ouamri, M.A.; Machter, Y.; Singh, D.; Alkama, D.; Li, X. Joint Energy Efficiency and Throughput Optimization for UAV-WPT Integrated Ground Network Using DDPG. *IEEE Commun. Lett.* **2023**, *27*, 3295–3299. [CrossRef]
46. Amhaz, A.; Elhattab, M.; Sharafeddine, S.; Assi, C. UAV-Assisted NOMA for Enhancing ISAC: A Deep Reinforcement Learning Solution. *IEEE Commun. Lett.* **2025**, *29*, 249–253. [CrossRef]
47. Fujimoto, S.; Hoof, H.; Meger, D. Addressing function approximation error in actor-critic methods. In Proceedings of the International Conference on Machine Learning (ICML), Stockholm, Sweden, 10–15 July 2018; pp. 1587–1596.
48. Puspitasari, A.A.; Lee, B.M. TD3 Algorithm-Based SWIPT With UAV-RIS Assistance for MIMO Communication. *IEEE Trans. Veh. Technol.* **2025**, *74*, 6284–6293. [CrossRef]
49. Hochreiter, S.; Schmidhuber, J. Long Short-Term Memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef]
50. Qin, J.; Zhu, M.; Pan, Z.; Li, Y.; Li, Y. Memory-based deep reinforcement learning for cognitive radar target tracking waveform resource management. *IET Radar Sonar Navig.* **2023**, *17*, 1822–1836. [CrossRef]
51. Meng, L.; Gorbet, R.; Kulić, D. Memory-based Deep Reinforcement Learning for POMDPs. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Prague, Czech Republic, 27 September–1 October 2021; pp. 5619–5626. [CrossRef]

52. Tang, B.; Liang, J.; Cai, Z.; Cai, T.; Zhou, X.; Chen, Y. MDRL-IR: Incentive Routing for Blockchain Scalability With Memory-Based Deep Reinforcement Learning. *IEEE Trans. Serv. Comput.* **2023**, *16*, 4375–4388. [CrossRef]
53. Kang, H.; Chang, X.; Mišić, J.; Mišić, V.B.; Fan, J.; Liu, Y. Cooperative UAV Resource Allocation and Task Offloading in Hierarchical Aerial Computing Systems: A MAPPO-Based Approach. *IEEE Internet Things J.* **2023**, *10*, 10497–10509. [CrossRef]