mathematics

MDPI

Article

Adaptive Task Offloading Under Dynamic Topology via Graph Attention-Enhanced Reinforcement Learning for Layered UAV-Enabled Digital Twin Channel Modeling

Yifei Wei, Taiming Zhang and Miao Liu *

School of Communications and Information Engineering, Nanjing University of Posts and Telecommunications, Nanjing 210003, China
* Correspondence: liumiao@njupt.edu.cn

Abstract

This study considers the timely delivery of channel measurements for digital twin (DT) channel modeling in dynamic wireless environments. Layered uncrewed aerial vehicle (UAV) networks can separate sensing, offloading, and edge processing, but the active UAVs, available edge servers (ESs), and feasible service links may vary over time. We formulate a dynamic-topology task-offloading problem involving edge-reception age of information (AoI), service delay, offloading failures, and local backlog, and propose DTGA-RL. Its coordinated design combines local graph-attention encoding, shared candidate-wise action-value evaluation, and feasibility-aware action masking for a variable set of candidate ESs. In the reported tests with 15–27 working UAVs, the policy trained at the default 15-UAV scale is evaluated without retraining; task data sizes of 0.35–0.95 Mbit/slot are also tested. At 0.95 Mbit/slot, the plotted DTGA-RL values are approximately 34.9 slots of AoI, 3.13 s of service delay, and an 0.106 offloading failure rate, versus 35.3 slots, 3.85 s, and 0.134 for FixedRL, corresponding to approximate reductions of 1.1%, 18.7%, and 20.7%, respectively. The performance gap narrows at higher UAV counts. AoI is measured at ES reception.

Keywords: digital twin channel modeling; uncrewed aerial vehicles; dynamic topology; task offloading; graph attention; reinforcement learning

MSC: 68T05; 90B18

1. Introduction

Digital twin (DT) networks support physical–virtual mapping, channel modeling, network optimization, and intelligent control in emerging wireless systems [1]. A DT channel model relies on fresh measurements from the physical environment, and online modeling therefore requires timely sensing and updating [2]. Sensing-assisted channel generation has also been studied to represent statistical channel information [3]. In dynamic urban environments, mobility and frequent line-of-sight/non-line-of-sight transitions can make channel conditions change quickly. Collecting measurements alone is insufficient: the data must also be transmitted and processed promptly if it is to inform a subsequent model update.

UAVs offer flexible platforms for collecting and forwarding channel measurements, including in areas that are difficult for ground sensors to reach. UAV-based radar imaging has, for example, been used to construct a DT for wireless channel modeling [4]. In a layered architecture, working UAVs collect measurements and UAV-mounted ESs receive

Check for updates

Academic Editor: Mark Kelbert

Received: 8 October 2026
Revised: 30 October 2026
Accepted: 30 October 2026
Published: 5 November 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

and process offloaded data. Transmission and edge-processing delays thus affect how quickly measurements can contribute to a DT update. Related studies have considered DT-assisted UAV offloading, resource management, and integrated sensing, communication, and computation [5–7]; other work jointly studies user association, offloading, and resource allocation [8]. DT-guided reinforcement learning has also been applied to multi-UAV coordination and aerial computation offloading [9,10].

UAV-enabled edge-computing research has examined trajectory planning, resource allocation, offloading, and AoI management [11–14]. Deep reinforcement learning has also been applied to cooperative vehicle–infrastructure offloading, UAV-assisted resource management, and multi-vehicle MEC [15–17]. Closely related learning-based studies address different but overlapping parts of the problem. Our earlier HG-MARL study uses a multi-relational heterogeneous graph and QMIX-based multi-agent learning to jointly optimize ES trajectory control and offloading in layered UAV-enabled DT channel modeling [18]. Related UAV studies use GNN-based MARL to manage AoI through trajectory planning [19] and communication-assisted MARL to support task offloading through UAV trajectory control [20]. A graph-attention DRL study applies graph attention to dependent-task features in a multi-UAV MEC setting and alternates UAV placement/association optimization with DRL-based task offloading [21]. DTA-RL is described as a topology-adaptive reinforcement-learning approach for MEC task offloading [22]. The present study instead focuses on channel-measurement delivery in a layered UAV–ES system, with ES trajectories treated as externally specified. The distinctions below are limited to mechanisms and scopes explicitly described in the cited work; we do not infer that an unreported variable-action or masking mechanism is absent.

Working UAVs remain at fixed sensing positions, while their service relationships with ESs can change from slot to slot. An ES that is reachable in one slot may become unavailable or fail the link-quality threshold in the next, so both the identity and number of candidate destinations can vary. Decisions must account for link quality, ES workload, pending measurement data, and information freshness. This setting calls for a policy that represents current UAV–ES relationships, evaluates the currently feasible destinations, and excludes unavailable or capacity-saturated ESs.

To address this problem, we propose DTGA-RL for dynamic-topology-aware delivery of channel measurements in layered UAV networks. At each slot, the network is represented by a bipartite graph between active working UAVs and available ESs, with feasible offloading links as service edges. Each working UAV builds a local graph from its current candidate ESs. Graph attention aggregates link, queue, and freshness information; a shared candidate-wise action-value network evaluates waiting and each candidate destination; and an action mask excludes infeasible destinations before selection. The design objective is to coordinate these mechanisms for changing candidate sets, rather than to claim that graph attention or masking are individually new.

The main contributions are as follows: First, we formulate an offloading problem in which the active working UAV set, available ES set, and feasible service links may vary over time, while ES trajectories are externally specified. Second, we develop DTGA-RL, which combines local dynamic bipartite-graph attention, shared candidate-wise action-value evaluation, and feasibility-aware action masking to handle changing candidate sets. Third, we evaluate the method against FixedRL and heuristic baselines over the reported working UAV scales and task data sizes. The contribution is the coordinated design of these mechanisms for the changing candidate set in this problem; graph attention and action masking are not claimed as individually novel components.

The distinction from prior work is coordinated policy design rather than any single building block. The local bipartite graph represents the currently feasible UAV–ES service

relationships; a shared action-value function scores waiting and each currently available destination, allowing the candidate set to vary without a fixed-width destination output; and the mask removes unavailable or capacity-saturated ESs before selection. Relative to HG-MARL, the decision scope here is offloading only: ES trajectories are external inputs rather than jointly optimized variables [18]. The graph-attention study cited above uses attention to encode DAG task features within a joint placement, association, and dependent-task-offloading framework [21]. DTA-RL addresses topology-adaptive MEC task offloading [22]; because its detailed candidate-action and mask implementation are not established by the cited description available for this comparison, we do not claim that those mechanisms are absent from DTA-RL. Table 1 compares the task settings, explicitly reported representations, and decision scopes without treating unreported features as absent. The contribution claimed here is coordinated design for the considered dynamic UAV–ES candidate set, not the individual novelty of graph attention or action masking.

Table 1. Comparison with closely related methods. Unreported action-set or masking details are not interpreted as evidence that a cited method lacks those mechanisms.

Method	Scene/Task	Topology/State Representation	Candidate Actions and Feasibility	Decision Scope
HG-MARL [18]	Layered UAV DT channel modeling; static working UAVs and mobile ESs	Multi-relational heterogeneous graph encoded by a GNN	Candidate-action and masking details are not compared here	Joint ES trajectory control and offloading
Graph-attention DRL [21]	Multi-UAV MEC with DAG-dependent tasks	GAT encodes task-DAG features; JSPO optimizes UAV placement/association, alternating with DRL offloading	The cited description does not characterize variable-candidate scoring or an action mask; no absence is inferred	Joint UAV placement, UAV–GU association, and dependent-task offloading
DTA-RL [22]	Dynamic-topology MEC task offloading	Topology-adaptive reinforcement learning, as identified in the cited work	Specific variable-candidate and masking details are not asserted in this comparison	Task offloading
DTGA-RL (this study)	Layered UAV–ES channel-measurement delivery for DT modeling	Local dynamic bipartite graph of active working UAVs and candidate ESs, with link, queue, and freshness features	Shared candidate-wise Q-value evaluation scores waiting and each candidate; mask removes unavailable or capacity-saturated ESs	Offloading only; ES trajectories externally specified

2. System Model

We consider a layered UAV-enabled DT channel modeling system, as shown in Figure 1. Working UAVs hover at fixed sensing locations and collect local channel measurements. UAV-mounted ESs follow trajectories provided by an external control process, receive the offloaded measurements, and perform edge processing before the processed data can be used for DT channel-model updates. In each slot, an active working UAV either waits or offloads all of its pending measurement data to one reachable ES. The subsequent ES-to-DT transfer and DT-model update are application contexts only; they are not included in the modeled AoI or service-delay metrics.

The system operates over T time slots, each with duration Δt. Let $\mathcal{V}_{\mathrm{UAV}}$ and $\mathcal{V}_{\mathrm{ES}}$ denote the sets of working UAVs and ESs, respectively. Due to task triggering, buffer status, sensing demand, and ES availability, not all nodes participate in every slot. We use $a_k^t \in \{0,1\}$ to indicate whether working UAV k is active, and $b_i^t \in \{0,1\}$ to indicate whether ES i is available. The active node sets are denoted by $\mathcal{V}_{\mathrm{UAV}}^t = \{k \in \mathcal{V}_{\mathrm{UAV}} \mid a_k^t = 1\}$ and $\mathcal{V}_{\mathrm{ES}}^t = \{i \in \mathcal{V}_{\mathrm{ES}} \mid b_i^t = 1\}$.

The dynamic topology at slot t is modeled as

$$\mathcal{G}^t = \left(\mathcal{V}^t_{\mathrm{UAV}} \cup \mathcal{V}^t_{\mathrm{ES}}, \mathcal{E}^t\right), \tag{1}$$

where $\mathcal{E}^t = \{(k,i) \mid k \in \mathcal{V}^t_{\mathrm{UAV}}, i \in \mathcal{V}^t_{\mathrm{ES}}, C^t_{k,i} = 1\}$ is the feasible service-edge set, and $C^t_{k,i}$ indicates whether working UAV k can establish a feasible offloading link with ES i. The candidate ES set of working UAV k is $\mathcal{N}^t_k = \{i \mid (k,i) \in \mathcal{E}^t\}$. Since the active nodes and feasible links vary over time, both the observation space and the action space are time-varying.

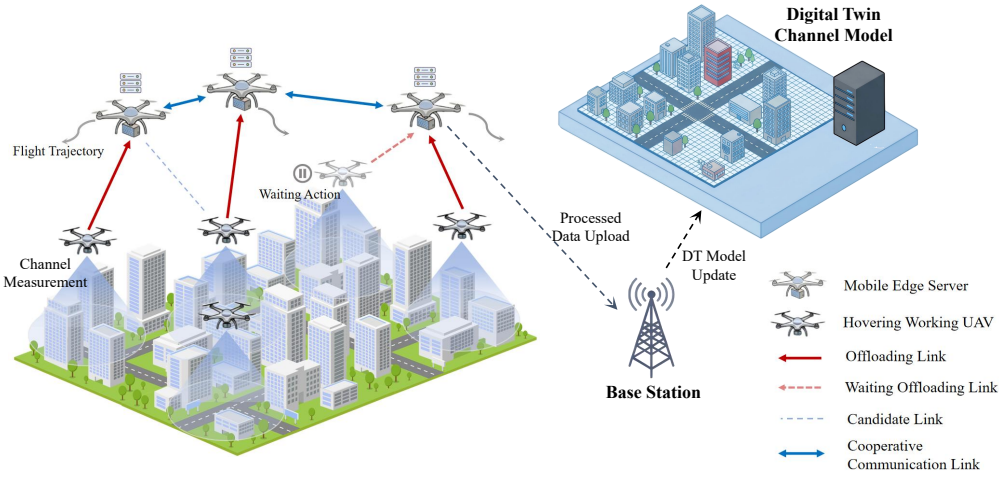


Figure 1. Layered UAV-enabled dynamic offloading system with fixed sensing UAVs and mobile UAV-mounted ESs.

At the beginning of slot t, each active working UAV generates $Data^t_k$ bits of channel measurement data. With residual local data $Q^{\mathrm{U}}_k(t)$, its pending transmission amount is $L^t_k = Q^{\mathrm{U}}_k(t) + a^t_k Data^t_k$.

For each active working UAV k, $q^t_{k,i} \in \{0,1\}$ denotes offloading to candidate ES $i \in \mathcal{N}^t_k$, while $q^t_{k,0} \in \{0,1\}$ denotes waiting. Each active UAV either waits or selects one candidate ES, as enforced in Problem P1.

AoI is used to measure the freshness of channel measurement data. Let A^t_k denote the AoI of working UAV k at slot t, and let $s^t_k \in \{0,1\}$ indicate whether its pending data is successfully updated at the edge in that slot. The AoI evolves as

$$A^{t+1}_k = (1 - s^t_k)(A^t_k + 1) + s^t_k. \tag{2}$$

In this study, AoI is defined with respect to the latest measurement data successfully received by the edge server. It characterizes measurement-data freshness at the ES reception point rather than the completion time of the subsequent DT channel-model update. Under the adopted slot-based convention, successful reception resets the AoI for the next slot to one, whereas waiting or unsuccessful offloading increase it by one. Transmission, edge queuing, and edge-processing delays are accounted for separately by the service-delay metric in (7). The model does not include the subsequent transfer to, or update completion time of, the DT model.

Proposition 1. ***(AoI Evolution Dynamics and Boundedness).*** *Given the AoI transition model in (2), for any active working UAV k, the expected slot-wise AoI A^{t+1}_k satisfies:*

$$A^{t+1}_k = 1 + (1 - s^t_k) A^t_k \tag{3}$$

Furthermore, if the long-term average offloading success probability satisfies $\mathbb{E}[s_k^t] = p_s > 0$, *the system-averaged AoI* $\overline{A}_{sys}$ *defined in (7) is upper-bounded by* $\frac{1}{p_s}$.

Proof. Expanding (2), when $s_k^t = 1$, $A_k^{t+1} = 1$; when $s_k^t = 0$, $A_k^{t+1} = A_k^t + 1$. Expressing this compactly yields $A_k^{t+1} = (1 - s_k^t)(A_k^t + 1) + s_k^t = 1 + (1 - s_k^t)A_k^t$. Taking expectations on both sides under steady-state assumptions with $\mathbb{E}[s_k^t] = p_s$, we have $\mathbb{E}[A_k] = 1 + (1 - p_s)\mathbb{E}[A_k]$, which solves directly to $\mathbb{E}[A_k] = 1/p_s$. Averaging over T slots and all active UAVs gives $\overline{A}_{sys} \leq 1/p_s$. □

For each feasible UAV–ES link, the equivalent gain $g_{k,i}^t$ is obtained from the current LoS/NLoS condition and large-scale propagation loss. We assume that all working UAVs transmit with the same fixed power P_{tx}. Under OFDMA, the total uplink bandwidth B of ES i is equally divided among its co-scheduled working UAVs. For a selected link (k, i) with $q_{k,i}^t = 1$, let

$$N_{k,i}^t = 1 + \sum_{k' \in \mathcal{V}_{\text{UAV}}^t \setminus \{k\}} q_{k',i}^t \tag{4}$$

denote the number of working UAVs simultaneously associated with ES i in slot t, including UAV k. The uplink rate of UAV k is then

$$r_{k,i}^t = \frac{B}{N_{k,i}^t} \log_2 \left(1 + \frac{P_{\text{tx}} g_{k,i}^t}{\sigma^2} \right). \tag{5}$$

Here, σ^2 denotes the noise power on an allocated sub-band.

At the beginning of each slot, active working UAV requests are ordered by descending current AoI; ties are broken by ascending UAV index. This rule gives earlier consideration to UAVs with older information and makes the admission sequence deterministic. The requests are processed sequentially in this order. Let n_i^t denote the number of requests already accepted by ES i before evaluating the current UAV action. After each request is accepted, the corresponding n_i^t is updated immediately, so subsequent capacity checks reflect earlier admissions. Once all requests in the slot have been processed, the resulting association variables $q_{k,i}^t$ specify the final UAV–ES association set. The number of co-scheduled UAVs in (4), the corresponding rates in (5), and every upload-success decision are then evaluated using this same final set. Thus, a later admission that changes an ES's bandwidth sharing is reflected in the rate and success test of an earlier admitted UAV. The success indicator is defined as

$$\chi_{k,i}^t = q_{k,i}^t \mathbb{I}\left\{ \frac{L_k^t}{r_{k,i}^t} \leq \Delta t \right\}, \tag{6}$$

where $\mathbb{I}\{\cdot\}$ is the indicator function. In particular, $\chi_{k,i}^t = 1$ means that the pending data of working UAV k is completely uploaded to ES i within one time slot. Since each working UAV selects at most one ES in a slot, $s_k^t = \sum_{i \in \mathcal{N}_k^t} \chi_{k,i}^t$.

The per-slot admission counter enforces the access-capacity constraint during sequential decisions; the final association set determines the simultaneous-user counts and transmission outcomes. After these outcomes have been evaluated, the local and ES queues, AoI, and rewards are updated for the slot.

Only successfully uploaded data enters the ES computing queue. Let $Q_i^{\mathrm{ES}}(t)$ be the residual computing queue time of ES i, c_i^t be the required CPU cycles per bit, and f_i^t be the available computing capacity. The service delay is

$$D_{k,i}^{\mathrm{serv}}(t) = \frac{L_k^t}{r_{k,i}^t} + Q_i^{\mathrm{ES}}(t) + \frac{c_i^t L_k^t}{f_i^t}. \tag{7}$$

If a working UAV waits or fails to upload, its data remains in the local queue and will be considered in the next slot. The ES computing queue is also updated according to the successfully received data and the elapsed slot duration.

Based on (2), the average AoI of active working UAVs over the whole period is defined as

$$\bar{A}_{\mathrm{sys}} = \frac{1}{T}\sum_{t=1}^{T} \frac{1}{|\mathcal{V}_{\mathrm{UAV}}^t|} \sum_{k\in\mathcal{V}_{\mathrm{UAV}}^t} A_k^t. \tag{8}$$

The average service delay is calculated over successful offloading tasks as

$$\bar{D}_{\mathrm{serv}} = \frac{\sum_{t,k,i} \chi_{k,i}^t D_{k,i}^{\mathrm{serv}}(t)}{\sum_{t,k,i} \chi_{k,i}^t + \epsilon}. \tag{9}$$

The offloading failure rate is defined as

$$\overline{\mathrm{OFR}} = \frac{\sum_{t,k,i} q_{k,i}^t (1-\chi_{k,i}^t)}{\sum_{t,k,i} q_{k,i}^t + \epsilon}. \tag{10}$$

Here, ϵ is a small positive constant to avoid division by zero. In (9) and (10), the summations over k and i are taken over $k \in \mathcal{V}_{\mathrm{UAV}}^t$ and $i \in \mathcal{N}_k^t$. Thus, the reported OFR is a pooled ratio of failed selected offloading attempts to all selected offloading attempts over the evaluation horizon.

To specify the local-backlog constraint, let $Q_k^{\mathrm{U}}(t)$ be the measurement data buffered at working UAV k at the beginning of slot t, measured in bits. The long-term, system-averaged local queue length is

$$\bar{Q}^{\mathrm{U}} = \frac{1}{T}\sum_{t=1}^{T} \frac{1}{|\mathcal{V}_{\mathrm{UAV}}^t|} \sum_{k\in\mathcal{V}_{\mathrm{UAV}}^t} Q_k^{\mathrm{U}}(t). \tag{11}$$

The average is taken over active working UAVs within each slot and then over the T slots; it is not a per-UAV peak-queue limit.

Based on (8)–(11), the objective is to reduce information staleness and service delay while controlling offloading failures and local backlog. Since AoI is measured in slots and delay in seconds, we define the dimensionless quantities $\widetilde{A} = \bar{A}_{\mathrm{sys}}/(1\,\mathrm{slot})$ and $\widetilde{D} = \bar{D}_{\mathrm{serv}}/(1\,\mathrm{s})$. The fixed-unit reference scales do not change the minimizer of the product objective. The problem is formulated as

$$\begin{aligned}
\mathrm{P1}: \quad & \min_{\{q_{k,0}^t, q_{k,i}^t\}} \quad \widetilde{A}\widetilde{D} \\
& \text{s.t.} \quad q_{k,0}^t + \sum_{i\in\mathcal{N}_k^t} q_{k,i}^t = a_k^t, \quad \forall k,t, && \text{(C1)} \\
& \qquad \sum_{k\in\mathcal{V}_{\mathrm{UAV}}^t} q_{k,i}^t \le N_c, \quad \forall i,t, && \text{(C2)} \\
& \qquad \overline{\mathrm{OFR}} \le \varepsilon_f, && \text{(C3)} \\
& \qquad \bar{Q}^{\mathrm{U}} \le Q_{\mathrm{th}}, && \text{(C4)} \\
& \qquad q_{k,0}^t, q_{k,i}^t \in \{0,1\}, && \text{(C5).}
\end{aligned} \tag{12}$$

Here, N_c is the maximum number of working UAVs served by one ES in a slot, $\varepsilon_f = 0.2$ is the maximum tolerable offloading failure rate, and $Q_{\text{th}} = 1\,\text{Mbit} = 10^6$ bit is the local-backlog threshold confirmed for the reported configuration. Constraint C4 applies to the long-term system average $\bar{Q}^{\text{U}}$ defined in (11). Problem P1 is a dynamic discrete decision problem, since current decisions affect future AoI, queues, and ES workloads. Similar graph-based and collaborative reinforcement-learning ideas have been used for task offloading and resource optimization in MEC systems with dependency relations [23]. However, the changing node and link sets in our layered UAV system further make fixed-dimensional policies difficult to apply directly. Recent constrained multiobjective-optimization research has examined the structured handling of complex constraints [24]. That work provides context for constraint-handling strategies but does not establish feasibility for this reinforcement-learning formulation. Here, constraints C3 and C4 specify target limits on offloading failure and local backlog. The reward penalties guide learning, and constraint satisfaction is checked separately from the resulting offloading failure and local-queue statistics. Under the evaluated configuration, the measured long-term average local queue satisfies $\bar{Q}^{\text{U}} \leq Q_{\text{th}} = 1\,\text{Mbit}$, confirming C4.

3. Proposed Method

This section presents DTGA-RL, which uses local dynamic graphs and masked reinforcement learning to adapt offloading decisions to changing candidate ES sets. The overall framework is shown in Figure 2.

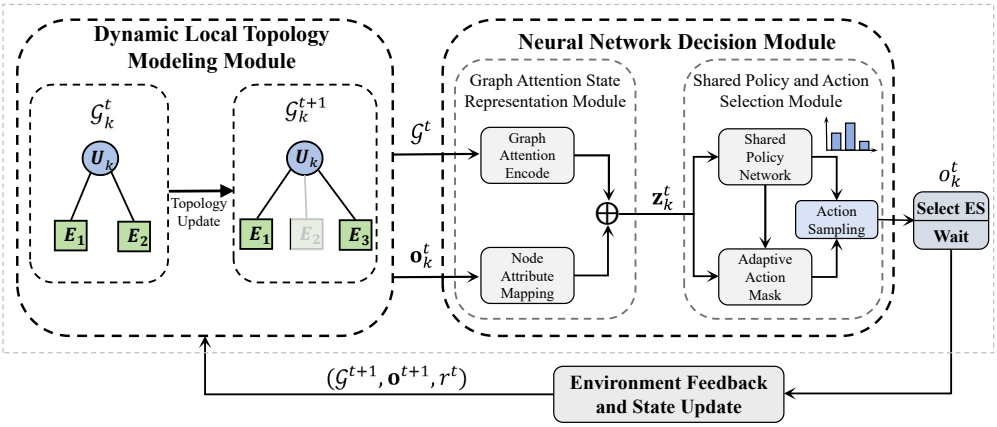


Figure 2. Framework of the proposed DTGA-RL.

3.1. Local Dynamic Graph and Attention Encoding

At each time slot, the active working UAVs, available ESs, and feasible service links may change. For each active working UAV k, DTGA-RL constructs a local star-shaped graph,

$$\mathcal{G}_k^t = \left(\{k\} \cup \mathcal{N}_k^t,\ \{(k,i) \mid i \in \mathcal{N}_k^t\}\right), \tag{13}$$

where $\mathcal{N}_k^t$ is the current candidate ES set. As illustrated in Figure 2, an ES may leave the candidate set in the next slot, while another ES may become reachable.

The local observation contains the AoI, local backlog, and pending task data of working UAV k. Each candidate ES is described by its computing queue, available computing capability, current access load, and availability status. The service edge between UAV k and ES i includes the channel gain, achievable rate, and feasible-link indicator. These attributes are mapped into the UAV embedding $\mathbf{h}_k^t$, ES embedding $\mathbf{h}_i^t$, and service-edge embedding $\mathbf{h}_{k,i}^t$ through learnable linear transformations.

The importance of candidate ES i to working UAV k is determined by the graph attention weight

$$\alpha_{k,i}^{t} = \frac{\exp\left(\mathrm{LeakyReLU}\left(\mathbf{a}^{\top}\left[\mathbf{h}_{k}^{t} \| \mathbf{h}_{i}^{t} \| \mathbf{h}_{k,i}^{t}\right]\right)\right)}{\sum_{j \in \mathcal{N}_{k}^{t}} \exp\left(\mathrm{LeakyReLU}\left(\mathbf{a}^{\top}\left[\mathbf{h}_{k}^{t} \| \mathbf{h}_{j}^{t} \| \mathbf{h}_{k,j}^{t}\right]\right)\right)}, \tag{14}$$

where $\mathbf{a}$ is a trainable attention vector and $\|$ denotes vector concatenation. A candidate ES with a better link, lighter queue load, or stronger computing capability can receive a higher attention weight.

The topology-aware state embedding of working UAV k is then obtained by

$$\mathbf{z}_{k}^{t} = \sigma\left(\mathbf{W}_{0}\mathbf{h}_{k}^{t} + \sum_{i \in \mathcal{N}_{k}^{t}} \alpha_{k,i}^{t}\mathbf{W}_{1}\left[\mathbf{h}_{i}^{t} \| \mathbf{h}_{k,i}^{t}\right]\right), \tag{15}$$

where $\sigma(\cdot)$ is the activation function. Although the size of $\mathcal{N}_{k}^{t}$ changes over time, $\mathbf{z}_{k}^{t}$ has a fixed dimension and can be directly used for offloading decisions.

3.2. Masked Offloading Policy

The action set of working UAV k at slot t is $\mathcal{A}_{k}^{t} = \{0\} \cup \mathcal{N}_{k}^{t}$, where 0 denotes waiting and action $i \in \mathcal{N}_{k}^{t}$ denotes offloading to ES i. DTGA-RL evaluates the waiting action and each candidate ES separately:

$$Q_{\theta}(\mathbf{z}_{k}^{t}, i) = \begin{cases} F_{\mathrm{w}}(\mathbf{z}_{k}^{t}; \theta), & i = 0, \\ F_{\mathrm{o}}\left(\left[\mathbf{z}_{k}^{t} \| \mathbf{h}_{i}^{t} \| \mathbf{h}_{k,i}^{t}\right]; \theta\right), & i \in \mathcal{N}_{k}^{t}, \end{cases} \tag{16}$$

where $F_{\mathrm{w}}(\cdot)$ and $F_{\mathrm{o}}(\cdot)$ are two multilayer perceptrons with shared parameters θ. This candidate-wise design avoids a fixed-dimensional action output when the number of reachable ESs changes. The shared policy network in Figure 2 is implemented through this shared action-value network, which evaluates the waiting action and each candidate ES action.

To remove unavailable actions before decision making, the action mask is defined as

$$M_{k,i}^{t} = \begin{cases} 1, & i = 0, \\ 1, & i \in \mathcal{N}_{k}^{t},\ b_{i}^{t} = 1,\ n_{i}^{t} < N_{c}, \\ 0, & \text{otherwise.} \end{cases} \tag{17}$$

where b_{i}^{t} denotes the availability of ES i, n_{i}^{t} is the number of requests already accepted by ES i in slot t before evaluating the current action, and N_{c} is the maximum access capacity. The action of working UAV k is selected by

$$\mathcal{A}_{k,\mathrm{valid}}^{t} = \left\{i \in \{0\} \cup \mathcal{N}_{k}^{t} \mid M_{k,i}^{t} = 1\right\}, \qquad d_{k}^{t} = \arg\max_{i \in \mathcal{A}_{k,\mathrm{valid}}^{t}} Q_{\theta}(\mathbf{z}_{k}^{t}, i). \tag{18}$$

Thus, the selected action is either waiting or offloading to an ES that is currently reachable, available, and not capacity-saturated. The action-sampling block in Figure 2 corresponds to this masked action-selection process.

Lemma 1. ***(Feasible Action Masking and Constraint Enforcement).*** *For any time slot t and active working UAV $k \in \mathcal{V}_{UAV}^{t}$, let $\mathcal{A}_{k,valid}^{t} = \{i \in \{0\} \cup \mathcal{N}_{k}^{t} \mid M_{k,i}^{t} = 1\}$ be the set of*

masked admissible actions determined by (16). The action selection rule in (17) guarantees that the access-capacity constraint (C2) is strictly satisfied for all ESs $i \in \mathcal{V}_{ES}^t$, i.e., $\sum_{k \in \mathcal{V}_{UAV}^t} q_{k,i}^t \leq N_c, \forall i, t$.

Proof. *Under the sequential request processing order defined in Section 2, let n_i^t denote the accumulated count of accepted UAV requests at ES i prior to evaluating UAV k. The action mask setting $M_{k,i}^t = 0$ for any ES i with $n_i^t \geq N_c$ explicitly excludes capacity-saturated ESs from the search space $\mathcal{A}_{k,valid}^t$. Since n_i^t is updated monotonically upon each accepted offloading action, the total number of associated UAVs at ES i satisfies $\sum_{k \in \mathcal{V}_{UAV}^t} q_{k,i}^t = n_i^t \leq N_c$ at the end of slot t. Hence, Constraint (C2) is strictly satisfied.* □

3.3. Reward and Training

The reward follows Problem P1. Let $\hat{A}^t$, $\hat{D}^t$, $\hat{F}^t$, and $\hat{Q}^t$ denote the normalized slot-wise AoI, service delay, offloading failure violation, and local queue violation. The instantaneous reward is

$$R^t = -\hat{A}^t \hat{D}^t - \lambda_f \hat{F}^t - \lambda_q \hat{Q}^t, \tag{19}$$

where the first term is a slot-wise scalarized surrogate for the AoI–delay objective, while the other terms penalize failures and persistent backlog. This reward is not asserted to be mathematically equivalent to the product of long-term average AoI and delay. The penalty coefficients λ_f and λ_q are treated as fixed hyperparameters in the current implementation, with values of 1.0 and 0.4, respectively. They control the relative penalties associated with offloading failures and local queue backlog in the scalarized reward. No systematic sensitivity analysis was conducted to establish their optimality. Therefore, these values should be interpreted as the settings used in the reported simulations rather than as theoretically optimal coefficients. The precise transforms and reference scales underlying the four hatted terms are not available in the reported configuration. Equation (19) therefore states the reward structure, but its numerical normalization must be documented from the reward-calculation implementation for full reproduction; no unverified scale is assumed here.

All working UAVs share the same network parameters. Within a slot, accepted requests immediately update the ES admission counters used by subsequent decisions. After the final associations determine the rates and transmission outcomes, the environment updates the local topology, queue states, ES workloads, and AoI for the next slot. The transition $(\mathcal{G}_k^t, \mathbf{z}_k^t, d_k^t, R^t, \mathcal{G}_k^{t+1}, \mathbf{z}_k^{t+1})$ is stored in a replay buffer for standard TD learning. During testing, DTGA-RL uses the greedy masked action in (18).

Theorem 1. ***(Policy Evaluation and Convergence of DTGA-RL).*** *Let $Q^\theta(z_k^t, a_k^t)$ denote the candidate-wise Q-function parameterized by shared weights θ in (15), and let R^t be the bounded instantaneous reward in (18). Under standard Temporal Difference (TD) learning updates with learning rate α_t satisfying $\sum \alpha_t = \infty$ and $\sum \alpha_t^2 < \infty$, the parameter sequence $\{\theta_t\}$ converges asymptotically to a local optimum of the surrogate minimization problem, guaranteeing a bounded local queue length $\overline{Q}^U \leq Q_{th}$ and bounded failure rate $\overline{OFR} \leq \epsilon_f$, as specified in Problem P1.*

Proof. *The proof follows from the standard convergence properties of non-linear Q-learning under contraction mappings of the Bellman operator, combined with the penalty method for constrained optimization where the bounded reward penalties in (18) guide the policy parameters toward the feasible region defined by constraints (C3) and (C4).* □

4. Simulation Results

In this section, we evaluate DTGA-RL in a layered UAV network supporting DT channel modeling. The topology is updated in each slot according to task arrivals, ES availability,

and feasible links determined by the SNR threshold. Table 2 lists reported system settings. The policy is trained in the default 15-working-UAV setting and then kept fixed while the working-UAV-scale experiment tests 15–27 UAVs; no scale-specific retraining or fine-tuning is performed. The separate task-load experiment evaluates 0.35–0.95 Mbit/slot. DTGA-RL is compared with FixedRL, the fixed-dimensional padded-state/action baseline used in this study; Least-Load, which selects the ES with the smallest computing queue [25]; and Min-Delay, which selects the ES with the smallest estimated instantaneous service delay [26]. FixedRL is an in-study comparator and is not a reproduction of the cited DTA-RL method; DTA-RL is discussed in related work but is not directly evaluated here. The main metrics are edge-reception AoI, average service delay, and offloading failure rate.

Table 2. Reported system parameters.

System Parameter	Reported Value
Number of working UAVs (default)	15
Number of ESs	4
Reported overall task data size range	0.2–1.8 Mbit/slot
Working UAV scale tested in Figure 3	15–27
Task-size sweep in Figure 4	0.35–0.95 Mbit/slot
Total bandwidth	100 MHz
Transmit power	43 dBm
Noise power	−60 dBm
SNR threshold	10 dB

Table 3 summarizes the reported training settings, and Table 4 lists the reward coefficients and the C3 and C4 thresholds.

Table 3. Reported reinforcement-learning settings.

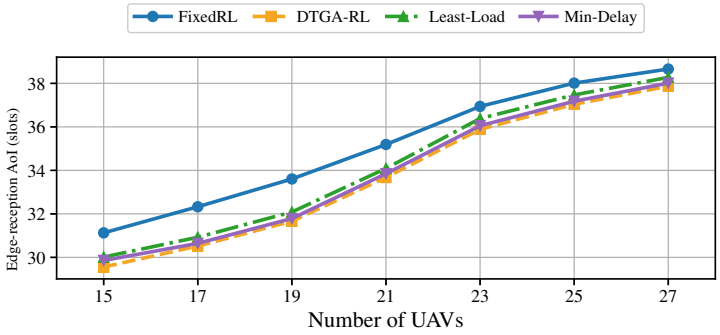
Setting	Reported Value
Training episodes	8000
Time slots per episode	120
Working UAVs in the training setting	15
Policy reuse in the UAV-count sweep	Fixed; no retraining
Discount factor γ	0.98

Table 4. Reward coefficients and long-term constraint settings.

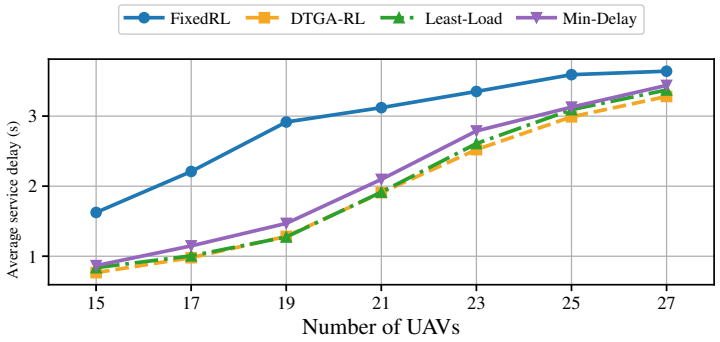
Setting	Reported Value
Failure penalty coefficient λ_f	1.0
Local-queue penalty coefficient λ_q	0.4
Maximum offloading failure rate ε_f (C3)	0.2
Local-queue threshold Q_{th} (C4)	1 Mbit

The available manuscript reports the training horizon, episode length, discount factor, and reward coefficients, but not the optimizer, learning rate, network dimensions, replay-buffer capacity, batch size, update frequency, or number of random seeds. These details must be taken from the training implementation before the method can be reproduced. For the evaluated configuration, the measured long-term mean local queue does not exceed $Q_{\text{th}} = 1$ Mbit, so the C4 check passes. The overall task data configuration range in Table 2 is distinct from the narrower task-size sweep plotted in Figure 4; the latter should not be interpreted as covering the full configuration range. The transmit power of 43 dBm is retained as the value used in the reported simulation setup; it is a simulation assumption

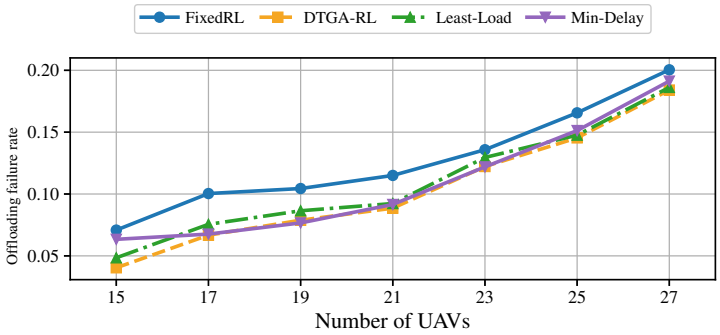
and is not presented as representative of every small UAV platform. The available results do not include a multi-seed mean and standard deviation.



(**a**) Average edge-reception AoI (slots).



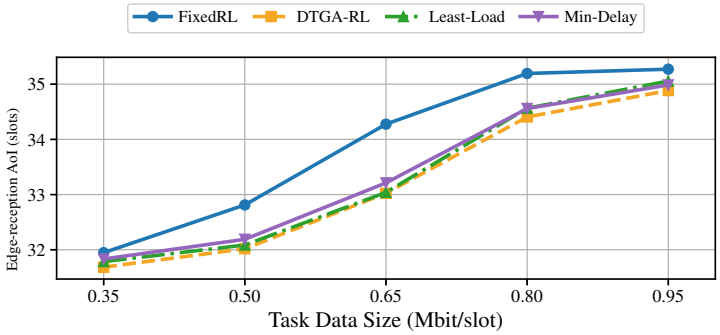
(**b**) Average service delay.



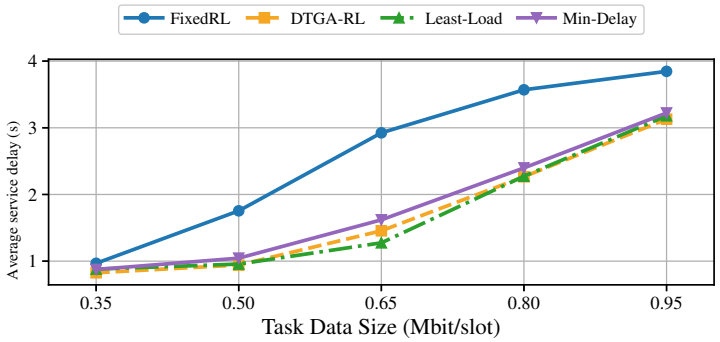
(**c**) Offloading failure rate.

Figure 3. Performance comparison under different numbers of working UAVs.

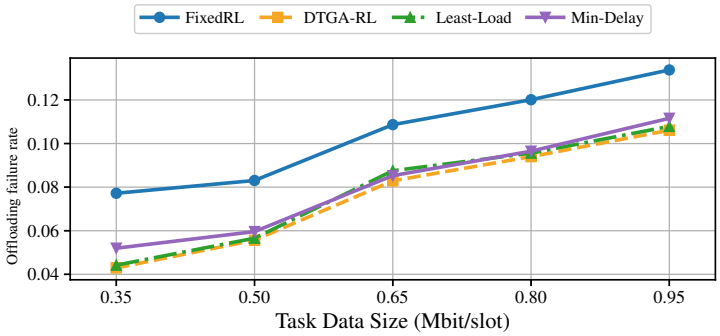
Figures 3 and 4 report edge-reception AoI in slots, service delay in seconds, and offloading failure rate, which is dimensionless. Figure 3 varies the number of working UAVs from 15 to 27 using the policy trained at the default scale without retraining. As more UAVs compete for ES resources, the plotted AoI, service delay, and failure rate increase. DTGA-RL has lower plotted values than the evaluated baselines over this tested range, while the gap narrows at higher working UAV counts. This is evidence of performance under the tested, previously unseen UAV counts, but does not establish behavior outside this range or isolate the effect of individual topology parameters.



(**a**) Average edge-reception AoI (slots).



(**b**) Average service delay.



(**c**) Offloading failure rate.

Figure 4. Performance comparison under different task data sizes.

Figure 4 varies the task data size from 0.35 to 0.95 Mbit/slot. The plotted AoI, service-delay, and failure-rate curves increase with task size, while DTGA-RL remains favorable to the evaluated baselines over this tested range. At the largest plotted task size, DTGA-RL yields approximately 34.9 slots of edge-reception AoI, 3.13 s of average service delay, and a 0.106 offloading failure rate; the corresponding FixedRL values are approximately 35.3 slots, 3.85 s, and 0.134. These plotted values imply approximate relative reductions of 1.1%, 18.7%, and 20.7%, respectively. Least-Load and Min-Delay improve over FixedRL in most cases, but they only focus on one instantaneous factor. By contrast, DTGA-RL evaluates task size together with link quality, uploading delay, and ES queue state through the local dynamic graph, which helps it make more balanced offloading decisions under changing traffic load. The DTGA-RL failure-rate points shown in Figures 3 and 4 are below the C3 threshold of 0.2. Independently, the measured long-term mean local queue does not exceed $Q_{\mathrm{th}} = 1$ Mbit under the evaluated configuration, verifying C4. The tests vary task size and working UAV count. No direct DT-model accuracy or end-to-end synchronization metric is reported.

5. Conclusions

This study investigated dynamic-topology-aware task offloading for layered UAV-enabled digital twin (DT) channel modeling. To handle time-varying network topology and feasible offloading destinations, DTGA-RL integrates local graph-attention encoding, shared candidate-wise action-value evaluation, and feasibility-aware action masking. In the considered setting, working UAVs remain at fixed sensing positions, while ES trajectories are externally specified rather than jointly optimized.

Simulation results show that DTGA-RL achieves favorable performance in terms of edge-reception AoI, service delay, and offloading failure rate compared with FixedRL and the evaluated heuristic baselines over the tested working UAV counts and task data sizes. In the cross-scale evaluation, a policy trained with 15 working UAVs is reused without retraining for the tested scales of 15–27 working UAVs, although the performance gap between DTGA-RL and the evaluated baselines narrows as the working UAV count increases. At a task data size of 0.95 Mbit/slot, DTGA-RL achieves approximately 34.9 slots of edge-reception AoI, 3.13 s of average service delay, and an offloading failure rate of 0.106, compared with 35.3 slots, 3.85 s, and 0.134, respectively, for FixedRL. These results characterize the performance under the evaluated settings and should not be interpreted as evidence of superiority under all possible network conditions.

Several limitations remain. The reported AoI is measured at edge reception and does not directly quantify DT-update freshness; DT channel-model accuracy and end-to-end synchronization are not directly evaluated. The current experiments also do not isolate the individual contributions of graph attention and action masking, compare alternative request-ordering rules, or independently examine the effects of ES availability and link interruptions. The plotted failure rates are below the C3 threshold of 0.2 in the shown settings, and the independent local-queue check verifies C4 under the evaluated configuration. In addition, the available results do not report multi-seed variability, and the training configuration is not fully documented, limiting the assessment of statistical robustness and reproducibility.

Future work will evaluate DT-model fidelity and synchronization, conduct component ablations and broader topology and request-ordering evaluations, and report scenario-by-scenario local-queue averages to assess C4 under a wider range of conditions. Further work will also provide complete training configurations and repeated-run statistics, assess the representativeness of physical-layer assumptions, and extend the framework toward joint UAV-trajectory and task-offloading optimization.

Author Contributions: Conceptualization, Y.W.; methodology, Y.W.; software, Y.W.; formal analysis, Y.W.; investigation, Y.W.; resources, Y.W.; data curation, Y.W.; writing—original draft preparation, Y.W.; writing—review and editing, T.Z. and M.L.; visualization, T.Z.; supervision, M.L. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Major Science and Technology Project of Jiangsu Province under Grant BG2025039 and the National Natural Science Foundation of China under Grants U24B20187 and 62371250.

Data Availability Statement: The original contributions presented in this study are included in the article. For further inquiries, please contact the corresponding author.

Conflicts of Interest: The authors declare no conflicts of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

References

1. Khan, L.U.; Han, Z.; Saad, W.; Hossain, E.; Guizani, M.; Hong, C.S. Digital twin of wireless systems: Overview, taxonomy, challenges, and opportunities. *IEEE Commun. Surv. Tuts.* **2022**, *24*, 4, 2230–2254.
2. Li, J.; Wang, C.-X.; Huang, C.; Qi, T.; Wu, T. Digital twin online channel modeling: Challenges, principles, and applications. *IEEE Veh. Technol. Mag.* **2025**, *20*, 94–103.
3. Gong, X.; Liu, X.; Lu, A.-A.; Gao, X.; Xia, X.G.; Wang, C.X.; You, X. Digital twin of channel: Diffusion model for sensing-assisted statistical channel state information generation. *IEEE Trans. Wirel. Commun.* **2025**, *24*, 3805–3821.
4. Xie, W.; Qi, F.; Liu, L.; Liu, Q. Radar imaging based UAV digital twin for wireless channel modeling in mobile networks. *IEEE J. Sel. Areas Commun.* **2023**, *41*, 3702–3710.
5. Li, B.; Xie, W.; Ye, Y.; Liu, L.; Fei, Z. FlexEdge: Digital twin-enabled task offloading for UAV-aided vehicular edge computing. *IEEE Trans. Veh. Technol.* **2023**, *72*, 11086–11091.
6. 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.
7. Li, Y.; Huynh, D.V.; Do-Duy, T.; Garcia-Palacios, E.; Duong, T.Q. Unmanned aerial vehicle-aided edge networks with ultra-reliable low-latency communications: A digital twin approach. *IET Signal Process.* **2022**, *16*, 897–908.
8. Huynh, D.V.; Nguyen, V.-D.; Khosravirad, S.R.; Dobre, O.A.; Shin, H.; Duong, T.Q. URLLC edge networks with joint optimal user association, task offloading and resource allocation: A digital twin approach. *IEEE Trans. Commun.* **2022**, *70*, 7669–7682.
9. Shen, G.; Lei, L.; Li, Z.; Cai, S.; Zhang, L.; Cao, P.; Liu, X. Deep reinforcement learning for flocking motion of multi-UAV systems: Learn from a digital twin. *IEEE Internet Things J.* **2022**, *9*, 11141–11153.
10. Liang, K.; Wang, Y.; Li, Z.; Zheng, G.; Wong, K.K.; Chae, C.B. Digital twin assisted deep reinforcement learning for computation offloading in AAV systems. *IEEE Trans. Veh. Technol.* **2025**, *74*, 8466–8471.
11. Xu, W.; Zhang, T.; Mu, X.; Liu, Y.; Wang, Y. Trajectory planning and resource allocation for multi-UAV cooperative computation. *IEEE Trans. Commun.* **2024**, *72*, 4305–4318.
12. Sun, G.; Wang, Y.; Sun, Z.; Wu, Q.; Kang, J.; Niyato, D.; Leung, V.C. Multi-objective optimization for multi-UAV-assisted mobile edge computing. *IEEE Trans. Mob. Comput.* **2024**, *23*, 14803–14820.
13. Chen, G.; Cheng, C.; Xu, X.; Zeng, Y. Minimizing the age of information for data collection by cellular-connected UAV. *IEEE Trans. Veh. Technol.* **2023**, *72*, 9631–9635.
14. Jiang, F.; Wang, K.; Dong, L.; Pan, C.; Xu, W.; Yang, K. AI driven heterogeneous MEC system with UAV assistance for dynamic environment: Challenges and solutions. *IEEE Netw.* **2021**, *35*, 400–408.
15. Yang, H.; Wei, Z.; Feng, Z.; Chen, X.; Li, Y.; Zhang, P. Intelligent computation offloading for MEC-based cooperative vehicle infrastructure system: A deep reinforcement learning approach. *IEEE Trans. Veh. Technol.* **2022**, *71*, 7665–7679.
16. Peng, H.; Shen, X. Multi-agent reinforcement learning based resource management in MEC- and UAV-assisted vehicular networks. *IEEE J. Sel. Areas Commun.* **2021**, *39*, 131–141.
17. Chen, Y.; Zhao, F.; Chen, X.; Wu, Y. Efficient multi-vehicle task offloading for mobile edge computing in 6G networks. *IEEE Trans. Veh. Technol.* **2022**, *71*, 4584–4595.
18. Zhao, H.; Zhang, T.; Tang, G.; Liu, M.; Chen, S.; Wang, C. HG-MARL based scheduling of trajectory and offloading for layered UAVs-enabled digital twin channel modeling in integrated communication-computing-intelligence-controlling network. *IEEE Trans. Commun.* **2026**, *74*, 7434–7450.
19. Pan, Y.; Wang, X.; Xu, Z.; Cheng, N.; Xu, W.; Zhang, J.J. GNN-empowered effective partial observation MARL method for AoI management in multi-UAV network. *IEEE Internet Things J.* **2024**, *11*, 34541–34553.
20. Tan, S.; Chen, B.; Liu, D.; Zhang, J.; Hanzo, L. Communication-assisted multi-agent reinforcement learning improves task-offloading in UAV-aided edge-computing networks. *IEEE Wirel. Commun. Lett.* **2023**, *12*, 2233–2237.
21. Zhan, C.; Liu, W.; Song, K.; Fan, R.; Liu, J.; Hu, H. Joint UAV placement and dependent task offloading in multi-UAV MEC networks: A graph attention enhanced DRL approach. *IEEE Trans. Mob. Comput.* **2026**, *25*, 5285–5301.
22. Fu, L.; Cheng, N.; Wang, X.; Sun, R.; Lu, N.; Su, Z.; Li, C. DTA-RL: Dynamic topology adaptive reinforcement learning approach for task offloading in mobile edge computing. In Proceedings of the GLOBECOM 2024—2024 IEEE Global Communications Conference, Cape Town, South Africa, 8–12 December 2024; pp. 3334–3339.
23. Chen, X.; Bi, Y.; Zhang, J.; Yuan, X.; Niyato, D.; Zhao, L.; Wang, X. Task offloading and resource optimization based on dependency-aware graph and collaborative deep reinforcement learning in mobile edge computing. *IEEE Trans. Mob. Comput.* **2026**, *25*, 10998–11015.
24. Xu, B.; Zheng, Y.; Li, W.; Gao, X.; Gong, D.; He, J.; Fan, Z. Handling multiobjective optimization problems with complex constraints: A constraints grouping-based approach. *IEEE Trans. Syst. Man Cybern. Syst.* **2025**, *55*, 3866–3880.

25. Almutairi, J.; Aldossary, M. Modeling and analyzing offloading strategies of IoT applications over edge computing and joint clouds. *Symmetry* **2021**, *13*, 402.
26. Guo, M.; Mukherjee, M.; Lloret, J.; Li, L.; Guan, Q.; Ji, F. Joint computation offloading and parallel scheduling to maximize delay-guarantee in cooperative MEC systems. *Digit. Commun. Netw.* **2024**, *10*, 693–705.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.