

Article

DRL-Based Backbone SDN Control Methods in UAV-Assisted Networks for Computational Resource Efficiency

Inseok Song¹, Prohim Tam¹ , Seungwoo Kang¹, Seyha Ros¹ and Seokhoon Kim^{1,2,*}

¹ Department of Software Convergence, Soonchunhyang University, Asan 31538, Republic of Korea; sis5041@sch.ac.kr (I.S.); prohimtam@gmail.com (P.T.); oooksw12@sch.ac.kr (S.K.); seyha@sch.ac.kr (S.R.)

² Department of Computer Software Engineering, Soonchunhyang University, Asan 31538, Republic of Korea

* Correspondence: seokhoon@sch.ac.kr

Abstract: The limited coverage extension of mobile edge computing (MEC) necessitates exploring cooperation with unmanned aerial vehicles (UAV) to leverage advanced features for future computation-intensive and mission-critical applications. Moreover, the workflow for task offloading in software-defined networking (SDN)-enabled 5G is significant to tackle in UAV-MEC networks. In this paper, deep reinforcement learning (DRL) SDN control methods for improving computing resources are proposed. DRL-based SDN controller, termed DRL-SDNC, allocates computational resources, bandwidth, and storage based on task requirements, upper-bound tolerable delays, and network conditions, using the UAV system architecture for task exchange between MECs. DRL-SDNC configures rule installation based on state observations and agent evaluation indicators, such as network congestion, user equipment computational capabilities, and energy efficiency. This paper also proposes the training deep network architecture for the DRL-SDNC, enabling interactive and autonomous policy enforcement. The agent learns from the UAV-MEC environment through experience gathering and updates its parameters using optimization methods. DRL-SDNC collaboratively adjusts hyperparameters and network architecture to enhance learning efficiency. Compared with baseline schemes, simulation results demonstrate the effectiveness of the proposed approach in optimizing resource efficiency and achieving satisfied quality of service for efficient utilization of computing and communication resources in UAV-assisted networking environments.



Citation: Song, I.; Tam, P.; Kang, S.; Ros, S.; Kim, S. DRL-Based Backbone SDN Control Methods in UAV-Assisted Networks for Computational Resource Efficiency. *Electronics* **2023**, *12*, 2984. <https://doi.org/10.3390/electronics12132984>

Academic Editor: Yolanda

Blanco Fernández

Received: 9 June 2023

Revised: 4 July 2023

Accepted: 5 July 2023

Published: 6 July 2023



Copyright: © 2023 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 (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Mobile cloud computing (MCC) is a highly efficient system located in central cloud; however, the end user faces several computing inadequacies due to backbone congestion and offloading delays. Therefore, mobile edge computing (MEC) method is proposed which offers extraction by leveraging storage, computing, communication, and network capacities [1–3]. The integration of networking system and cloud/edge computing brings network cloudification which leverages existing cloud and edge computing infrastructure to host virtual network functions. The convergence of networking and cloud/edge computing necessitates a comprehensive perspective on the advancements and the delivery of composite network-cloud/edge services, which offers advantages, such as enhanced resource utilization, cost reduction, and emerging new prospects for stakeholders [4].

However, without mobility, the coverage extension is deficient, which further leads numerous researchers/organizations to opt for the cooperation between unmanned aerial vehicles (UAV) and MEC that would bring a variety of benefits. With the growing popularity and deployment of UAV in various applications, there is an increasing demand for efficient and reliable communication networks to support UAV-assisted operations [5,6]. UAV-assisted networks have the potential to enable a wide range of applications, including

aerial surveillance, disaster management, etc. [7–9]. Nevertheless, UAVs have several limitations in terms of energy, computation resources, etc. Moreover, the dynamic and unpredictable nature of UAV mobility poses significant challenges in ensuring seamless connectivity and resource efficiency in control entities, which requires a thorough proactive and autonomous model.

Software-defined networking (SDN) has emerged as a promising paradigm for managing and controlling network resources in a flexible and centralized manner, which can be abstracted from the UAV states for intelligent flow management [10–13]. The need for wireless communication infrastructure in critical applications is assisted by utilizing UAVs equipped with NFV/SDN capabilities, which are crucial technologies for enabling flexible control of UAV networks and computing [14]. By decoupling the control plane from the data plane, SDN allows for dynamic resource allocation and efficient network management [15]. However, the conventional SDN architectures are primarily designed for terrestrial networks and may not be well-suited for UAV-assisted networks due to the unique characteristics, such as mobility, limited energy, and limited processing capabilities. Therefore, intelligent agent modeling using deep learning or deep reinforcement learning (DRL) is suggested for integration [16,17].

In [18], authors present a state-specific environment on UAV observations, including SINR measurement, height of UAVs, and spectral efficiency, that offer insightful context into the agent on applied actions on (1) base station selection with transmission power settings, and (2) UAV elevation. The proposed system evaluates the action efficiency by rewarding joint maximization of energy and throughput until the converged policy is obtained. Furthermore, in [19], an analytical framework is proposed to assess the coverage probability in mobile networks assisted by UAVs, considering clustered users, varying UAV heights, and imperfect beam alignment. Figure 1 presents the overview interactions of unmanned aerial resource system (UARS) states gathering for agent training, action configuration, and policy installation in SDN controller (SDNC). With the advancement of intelligent modeling, the controlling agent can be generated in a flexible way using complete network state observation, action-based autonomous configuration, and prediction models [20–22]. By leveraging the capabilities of DRL, the proposed approach aims to enhance the resource efficiency and adaptability of SDN-based control mechanisms in UAV-assisted networks.

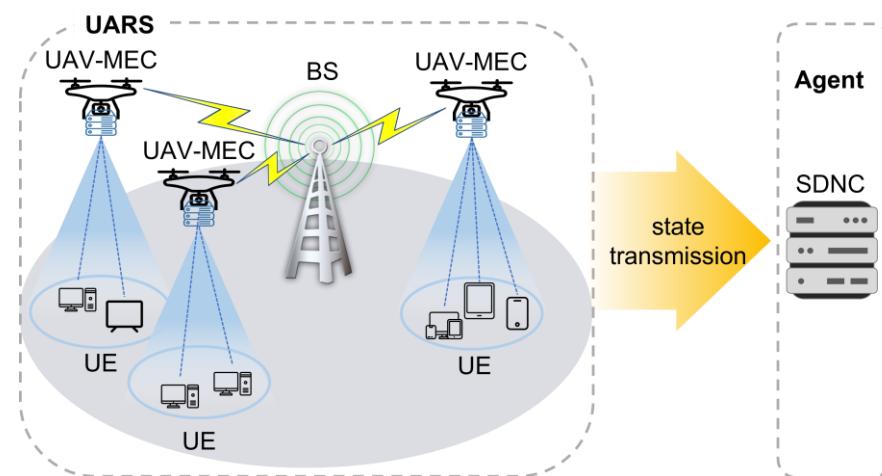


Figure 1. UARS environment for state gathering to interact with the agent controller.

The main contribution of this paper is the development of an agent with computational resource efficiency (CRE) objective function to optimize performance metrics, such as throughput, latency, and energy consumption in UAV-assisted networks. The proposed system utilizes DRL agents to learn the optimal control policies for handling computing

resource, appropriate node selection, and network management tasks based on various state observation features. Overall, we can highlight the following contributions:

- The working flow of the system architecture are given in this work by mentioning the main interfaces, such as (1) UE-to-UAV, (2) UAV-to-MEC, (3) UAV/MEC-to-SDN, (4) SDN-to-DRL, and (5) SDN, as a central controller to the whole network architecture. We provide insights into the feasibility and effectiveness of DRL in optimizing UAV states in dynamic and resource-constrained environments.
- The reward evaluation metrics are proposed to formulate the measurement of resource efficiency and adaptability in UAV-assisted networks through intelligent network management and SDN flow controls.
- We offer a comprehensive evaluation of the proposed system using OpenAI and network simulation software to highlight its advantages over existing solutions in several terms of resource utilization, latency, and energy scoring metrics.

2. Related Works

2.1. UAV-MEC for Resource Efficiency

The issue of energy-efficient resource allocation has been of note in modern networks, and has motivated researchers towards the implementation of UAV-enabled MEC systems. In [23], the authors aim to minimize the combined power consumption of UE and UAV. The minimization problem is well-formulated on total power consumption while considering latency and coverage constraints. The proposed algorithm addresses nonconvex problems through iterative optimization of (1) user association, (2) power control, (3) computation capacity allocation, and (4) location planning. In [24], a novel approach utilizing UAV in a MEC-enabled VANET is proposed to deliver low latency and reliable computing services to vehicles. The optimization objective focuses on minimizing comprehensive task processing delay by considering transmission models, security assurance, and task computation models. To optimize VANETs, the authors propose a network scheme that jointly considers these 3-tuple policies, and by leveraging the MEC-enabled UAV-assisted VANET architecture, the methods harness the communication capabilities of UAVs to enhance VANETs' computational abilities. An iterative algorithm based on the relax-and-rounding method and the Lagrangian method is used, and the simulation results capture the key performance metrics on successful task processing ratio and task processing delay. Furthermore, in [25], authors investigated the interaction between multiple users, drones, edge servers, and the cloud, aiming to optimize resource allocation and offloading decisions in a multi-user scenario. The minimization of energy consumption and delay is formulated to consider UAV offloading, edge server computing cost, and communication processing delay, in terms of local device computing power. The authors proposed MEC architecture for IoT devices, a drone-based edge computing layer, and edge computing server layer. The offloading strategy to minimize energy costs and maximum delay is formulated as a non-convex quadratic constrained quadratic program and a heuristic algorithm based on semi-definite relaxation and adaptive adjustment is used as a solver.

2.2. DRL for UAV-MEC Control

The autonomy of DRL can be appended as a solving agent in the UAV-MEC systems in terms of resource allocation, offloading decisions, and other optimization approaches. In [26], authors focus on a UAV-assisted wireless IoT network and introduces a multi-agent DRL framework combined with a round-robin resource-scheduling algorithm to optimize joint resource management. A system model is designed to capture the dynamic and heterogeneous nature of the environment, considering various constraints, such as user count, channel gains, noises, and power consumption. K-means and round-robin algorithm are used to efficiently handle service requests from IoT users in urban and sub-urban clusters. The proposed DRL framework is applied to optimize resource management for UAV-assisted IoT devices in the primary system model. In [27], an innovative application scenario where a UAV operates in a complex urban environment is considered. The UAV

receives computing tasks from clients within a specific flight range, leverages mobile edge computing servers for task processing, and optimizes its trajectory to enhance communication quality and improve the computing rate of the MEC network. By employing DRL and experience replay techniques to generate optimal offloading decisions and introducing a time frame allocation algorithm for resource allocation, the problem becomes predictive and is effectively solved [28,29]. The proposed scheme, combining UAV-assisted MEC networks, DRL, and radio map optimization, represents a novel and systematic approach to resource optimization. By integrating UAV and MEC, the accessibility and cost-effectiveness are reached, and by utilizing DRL, the computation offloading in a UAV-assisted MEC network can be optimized. The integrated approach can simplify the complexity of the original problem, resulting in improved computational and energy efficiencies with maximized computing rates [30].

2.3. DRL-Based SDN for Efficient Network Management

SDN enhances the scalability, flexibility, and control efficiencies of UAV-MEC networks by simplifying the addition or removal of UAVs and MEC servers, allowing traffic optimization/service provisioning, and enabling automated network orchestration, policy enforcement, and real-time monitoring, thereby enhancing operational efficiency in UAV-MEC networks [31]. To further advance SDNC in complex network environments, DRL has been integrated in several studies. In [32], SDN-based dynamic task scheduling and resource management approach using DRL for IoT traffic scheduling is proposed. The objective is to achieve high network performance by minimizing latency and ensuring energy efficiency. The proposed solution introduces an architectural design and formulates a task assignment, also called a scheduling problem. The approach offers effective trade-offs between response time constraints, model fidelity, inference accuracy, and task schedule-ability. Furthermore, in [33], an integrated intelligent algorithm in SDN-based QoS-routing optimization is proposed to enable more intelligent dynamic routing. Offline training methods are used for supervised learning-based models, while DRL-based models can be trained both online and offline. The study achieved intelligent online QoS-routing optimization solution using SDN and asynchronous advantage actor-critic, which introduces the algorithm for dynamic routing decisions and the DRL-enabled framework for real-time data collection and strategy learning.

3. DRL-Based Backbone SDN Control Methods in UAV-Assisted Networks

This section covers the system architecture, environment modeling, and algorithm designs with specified interfaces of the proposed framework. Throughout these subsections, the proposed framework is described overall by how the agent connects and operates for policy installation to achieve the objectives of computational resource efficiency.

3.1. System Architecture

The system architecture, as illustrated in Figure 2, emphasizes the relations of UAV-MEC in UARS with the base stations (BS), 5G service-based architectures (SBA), UAV access and mobility management function (*U-AMF*), and UAV session management function (*U-SMF*), and the interactions with the agent in SDNC through user plane function (UPF).

The workflow of task offloading in SDN-enabled 5G UAV-MEC networks starts from the task identification, where UE identifies computation-intensive tasks from resource-intensive applications that can be offloaded to the selected MEC for processing. Task offloading request is sent by UE to the *DRL-SDNC*, which includes the states on computational requirements, data size, and QoS constraints. *DRL-SDNC* receives the task offloading request and analyzes it for determining the appropriate network resources and MEC allocation for a complete/partial task processing. Comprehensively, *DRL-SDNC* allocates the necessary resources, including computational resources in MEC, bandwidth, and storage, based on the task requirements and network conditions. UPF is used to exchange tasks between MEC. *DRL-SDNC* configures the rule installation based on state observations

and the proposed agent evaluation indicators, such as network congestion, computational capabilities of the UE, and energy efficiency factors. Furthermore, *DRL-SDNC* provides the UE with the necessary information and interfaces to intelligently transmit the task data to the designated MEC for processing. Once the task processing is complete, SDNC notifies the UE about the completion status and provides the necessary interfaces for retrieving the processed outputs.

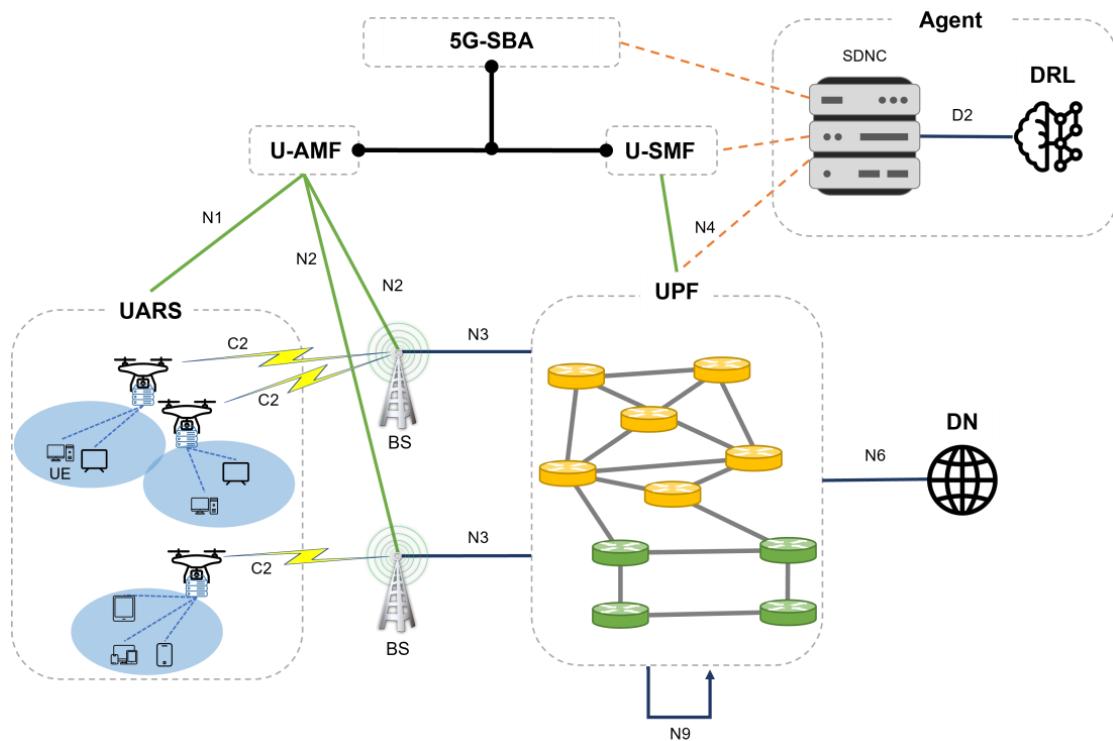


Figure 2. System architecture of the UAV-MEC environment in SDN-enabled 5G networks.

Through the access part of 5G UAV interfaces, *UARS* includes the radio interface between the UAV and the ground network, which enables wireless communication between the UAV and the 5G network infrastructure. *N1* interface covers the interaction between UE or UAV and the 5G access network by handling the wireless communication protocols and connectivity between the UE and the access network. *N2* interfaces cover the interactions between different 5G access network nodes by handling communication and coordination, such as base stations, in order to provide coverage and handover support for our proposed UAV-MEC. And for *C2* interface, it handles the UAV and the 5G control plane by allowing the control and management of the UAV's movements, configurations, and interactions with the 5G network.

N3 handles signaling and control messages related to mobility management, session establishment, and authentication. *N9* is the interface within UPF (e.g., SGW and PGW) which carries user data and controls messages for packet forwarding and routing purposes, which is later used for abstracting information and state observation to *DRL-SDNC*. *N6* and *N4* offer the interactions between UPF (e.g., PGW) and the data network (DN) or external control entities by handling the control plane signaling for mobility management and policy installation between the 5G core network and the proposed *DRL-SDNC*.

In our studies, the state gathering process and action configuration are important to collaborate with UPF, which includes the user plane data forwarding, routing, and optimization information, by performing tasks, such as packet filtering, forwarding, and traffic management. Moreover, 5G-SBA is also an entity to observe the states and enable the action deployment of flexible and scalable services in UAV-MEC environment. 5G-

SBA provides a service-based interface model for efficient communication and interaction between network functions and services.

For *U-AMF*, the agent leveraged the capability for UAV to manage access and mobility-related functions. And for *U-SMF*, the proposed system aims to handle the UAV session establishment, policy enforcement, and management of data flows between the UE, UAV, and the network infrastructure. These interfaces and entities work collaboratively for global UAV-assisted network state abstractions, environment connectivity, agent's action-based control, and data management in our proposed system architecture for ensuring reliable communication, and adequate computational resources between UE tasks and UAV-MEC selection/placement.

3.2. Environment Modeling

From the above-mentioned system architecture, the primary entities that generate states and configure the action rules must be modeled into softwarization for interacting and ordering the offloading policies into uses by the agent controller. Table 1 presents the important notations and its description used in this proposed DRL-based system.

Table 1. Key notations used in the environment initialization and the proposed agent interactions.

Notation	Description
$T = \{1, 2, \dots, t\}$	Set of timeslots
$I = \{1, 2, \dots, i\}$	Set of UAVs
$M = \{1, 2, \dots, m\}$	Set of MECs
$N = \{1, 2, \dots, n\}$	Set of UEs
$J = \{1, 2, \dots, j\}$	Set of offloading tasks
$S = \{1, 2, \dots, s\}$	Set of small base stations
$P_i^t(x_t, y_t, z_t)$	Position coordination (x -axis, y -axis, height) of UAV- i at timeslot- t
$D_i^t(d_{n,i}, d_{s,i})$ (e_i^t, e_n^t)	Distance metrics of UAV- i between UE- n and BS- s States of remaining energy metrics of UAV- i and UE- n
$(C_{\max,m}^t, C_{\text{re},m}^t)$	Maximum computing and remaining resources of MEC- m at timeslot- t
T_{bw}	Total system bandwidth
$(bw_{n \rightarrow i}^t, g_{n \rightarrow i}^t)$	States of bandwidth allocation and channel gain between UE-to-UAV
$(bw_{i \rightarrow s}^t, g_{i \rightarrow s}^t)$	States of bandwidth allocation and channel gain between UAV-to-BS
$(p_n^t, \forall \delta)$	Transmission power of the devices and overall noises
$\tau_n^t(\rho_n^t, \gamma_n^t, \mathbb{C}_n^t)$	States of tasks from UE- n at timeslot- t with 3-tuple information of task sizes, upper-bound tolerable delays, and computation workload
$R_{env}^t(r_d^t, r_e^t, r_{cr}^t)$	Primary reward evaluation metrics of the environment per episode, which formulating by 3 sub-rewards on joint delay, energy, and computational resource efficiency
$e_t(s_t, a_t, r_t, s_{t+1})$	Experience replay batches at t including the state, action, reward, and transited next-state.

The task process can be described in 4 primary phases as follows:

1. **Task offloading from UE to the selected UAV-MEC:** In each timeslot- t , UE- n can offload task- j to the integrated paring UAV-MEC ($i - m$) node. The task offloading decision is given by action output from the proposed agent, which can be presented as $a_i^{off}(t) = \{n_j, i, m\}$, indicating the index of UE- n which offloads a task- j to UAV-MEC system composed of UAV- i and MEC- m . The communication model, denoted as data rate $U_{n \rightarrow i}^t$, between UE-to-UAV is presented in Equation (1) which is associated with the states of allocating bandwidth $bw_{n \rightarrow i}^t$, channel gain $g_{n \rightarrow i}^t$, transmission power p_n^t ,

and overall noises $\forall\delta$, including the interference between UAVs and BSs and also additive white Gaussian noise.

$$U_{n-i}^t = bw_{n \rightarrow i}^t \log_2 \left(1 + \frac{p_n^t g_{n \rightarrow i}^t}{\forall\delta} \right) \quad (1)$$

2. **UAV-MEC task processing:** The integrated UAV-MEC system receives the offloaded tasks from UEs and performs the computing to get the expected task output. The task processing can be represented as $j_{i-m}^{comp}(t)$ indicating the processed task, consumed energy/resource, and executed times.
3. **Task offloading from UAV-MEC to BS:** The UAV-MEC system can further offload processed tasks to BS for transmission or communication purposes. In our environment, within small BS-s, there is a single UAV and single MEC which assists with the task computation. The communication model of UAV-to-BS is presented in Equation (2) by formulating uplink data rates associated with the states of allocating bandwidth $bw_{i \rightarrow s}^t$, channel gain $g_{i \rightarrow s}^t$, transmission power p_i^t , and overall noises/interference.

$$U_{i-s}^t = bw_{i \rightarrow s}^t \log_2 \left(1 + \frac{p_i^t g_{i \rightarrow s}^t}{\forall\delta} \right) \quad (2)$$

4. **Base Station task processing:** Base stations receive tasks from the UAV-MEC system and perform task processing or transmission. The state information from the tasks is gathered for the DRL-SDNC agent.

For DRL environment and agent initialization, the components can be described following Markov decision process, which primarily consists of states, actions, and reward, which are expressed in the following sub-sections, and in Equations (3)–(5), respectively.

3.2.1. States of UAV-MEC and Tasks from UE

- $P_i^t(x_t, y_t, z_t)$ represents the coordination of geographical UAV positions, and its trajectory P_i^{t+1} is set by the agent output in terms of distance metrics and offloading task patterns.
- $D_i^t(d_{n,i}, d_{s,i})$ represents the distance between UAV- i to local task owner, UE- n , and its pairing small BS-s.
- e_i^t and e_n^t represent the states of energy of UAV- i and UE- n at timeslot- t , respectively.
- Within paired MEC- m , the states of total computation workload and remaining resources at timeslot- t are denoted as $(C_{max,m}^t, C_{re,m}^t)$, respectively, for further re-allocations and enhancing decision-making policies.
- In communication model, the states of total bandwidth T_{bw} is observed from the environment. And the bandwidth allocation and channel gain between UE-to-UAV and UAV-to-BS, $(bw_{n \rightarrow i}^t, g_{n \rightarrow i}^t)$ and $(bw_{i \rightarrow s}^t, g_{i \rightarrow s}^t)$, are gathered, respectively.
- From local UE, the states of tasks and local computational resources are observed for formulating the offloading decision of local computation, and complete, or partial task offloading. τ_n^t represents the tasks from UE- n at timeslot- t , which consists of a 3-tuple information of task sizes, upper-bound tolerable delays, and computation workload, denoted as ρ_n^t , γ_n^t , and C_n^t , respectively.
- An experience tuple $j_{i-m}^{comp}(t)$ from processed tasks including the consumed energy/resource, and time spent from experiences.

$$s_t = \{P_i^t, D_i^t, e_i^t, e_n^t, C_{max,m}^t, C_{re,m}^t, T_{bw}, bw_{n \rightarrow i}^t, g_{n \rightarrow i}^t, bw_{i \rightarrow s}^t, g_{i \rightarrow s}^t, \tau_n^t, j_{i-m}^{comp}(t)\} \quad (3)$$

3.2.2. Actions of DRL-SDNC

The offloading decision-maker and flow scheduling, represented by $a_i^{off}(t)$, from the backbone SDN control methods are synchronized as an action from the agent, which have the global views of the UAV data plane and abstraction programmability. The mechanism of the DRL-SDNC interacts with the controllers for understanding the experience batches of

resource allocation and performance patterns in each observed state iteration. The actions also cover (1) the optimal MEC selection $a_{m|c}^{sel}(t)$ based on computational resource and (2) load balancing $a_{\forall m}^{bal}(t)$ over all the edge servers. SDN actions collaborates with the proposed agent by optimizing flow rule installations and alleviating heavy congestions for efficient task completion times within upper-bound tolerable delays.

$$a_t = \{a_i^{off}(t), a_{m|c}^{sel}(t), a_{\forall m}^{bal}(t)\} \quad (4)$$

3.2.3. Rewards and Evaluation on Optimal Policy Selection

The evaluation of the proposed agent and SDN control methods measures the efficiency of applying action a_t into UAV-MEC environment state s_t by obtaining reward r_t before transiting to the next state s_{t+1} . In our proposed method, the primary reward, denoted as R_{env}^t , is a complete formulation from sub-rewards on delay, energy, and computational resources, denoted as r_d^t , r_e^t , and r_{cr}^t , respectively. The critical weights ω on computational resources r_{cr}^t is adjusted for adapting partial of exceptional mandatory delay and energy requirements to primarily serve the efficiency of MEC computational resource placement and offloading decisions.

$$R_{env}^t = r_d^t + r_e^t + \omega r_{cr}^t \quad (5)$$

The optimal policy $\pi(t)^*$ is the SDN backbone control policy from the overall batches that maximize the long-term reward expectation by experiencing each rule condition, which is presented in (6). The value function formulates the expectation of increasing reward summation following the selected policy π from state s . Equation (7) presents the standard optimal q-value function following the bellman equation.

$$\pi(t)^* = \underset{\pi}{\operatorname{argmax}} \mathbb{E}_{\pi} \left[\sum_{t \in T} \gamma^t R_{env}^t \right] \quad (6)$$

$$Q^*(s, a) = \mathbb{E}_{\sim s'} \left[R_{env}^t + \gamma \underset{a'}{\operatorname{argmax}} Q^*(s', a') \right] \quad (7)$$

3.3. Algorithm Flows

The execution phases of this proposed control method can be trained through these four following phases:

1. **Hyperparameter/parameter initialization:** (1) Define the number of episodes, representing the iterations of the DRL training process. (2) Initialize the value function, which estimates the expected return for each state-action pair in the environment. (3) Initialize the policy function, which maps states to actions, and set the value of epsilon for epsilon-greedy exploration, with a decaying value over each iteration.
2. **State initialization:** This phase includes the capturing process from data plane context, such as the geographical positions of the UAVs, calculating the distance between UAV and the local task owner, recording the energy states of the UAVs, tracking the total computation workload and remaining resources, and evaluating the bandwidth allocation and channel gain between UE-to-UAV and UAV-to-BS. This information is stored as an experience tuple, as expressed in Equation (3).
3. **Iterative learning:** Within each episode, the system performs the eight following steps: (1) Select an action based on the current state using the policy function. Actions can include optimal MEC selection, load balancing across edge servers, and offloading decision-making and flow scheduling as $\{a_i^{off}(t), a_{m|c}^{sel}(t), a_{\forall m}^{bal}(t)\}$ tuples. (2) Execute the selected action in the environment and observe the resulting state. (3) Calculate the reward based on a complete formulation in Equation (5) that includes sub-rewards related to delay r_d^t , energy consumption r_e^t , and computational resources ωr_{cr}^t . (4) Up-

date the value function and policy function using the observed state, action, reward, and next state. (5) Store the experience tuple $e_t(s_t, a_t, r_t, s_{t+1})$ in a memory buffer for training the online and target networks (using per batch sampling). (6) Perform gradient descent optimization on the loss function to update the model's parameters. (7) Decay the value of epsilon to reduce exploration over each iteration. The above steps are repeated for the specified number of setting number of episodes from phase 1.

4. **Post-training completion:** The DRL model can be used for decision-making in real-time scenarios, considering the optimal learned policy for selecting optimal actions based on observed states. The context and improved accuracy of policy selection will be input into SDNC for flow installation and rule settings.

Figure 3 presents a single-looped schematic flows starting from (1) initializing the states from Equation (3), (2) selecting action (whether explored or exploited) from Equation (4), (3) calculating the total rewards from Equation (5), (4) formulating the transition probability to next state, (5) storing the experience batches which includes $e_t(s_t, a_t, r_t, s_{t+1})$, (6) feeding (s_t, a_t, r_t) to train the online network, and (s_{t+1}) to the target network, also exchanging weights for network improvement, and finally (7) obtaining the output as the recommended next action a_{t+1} .

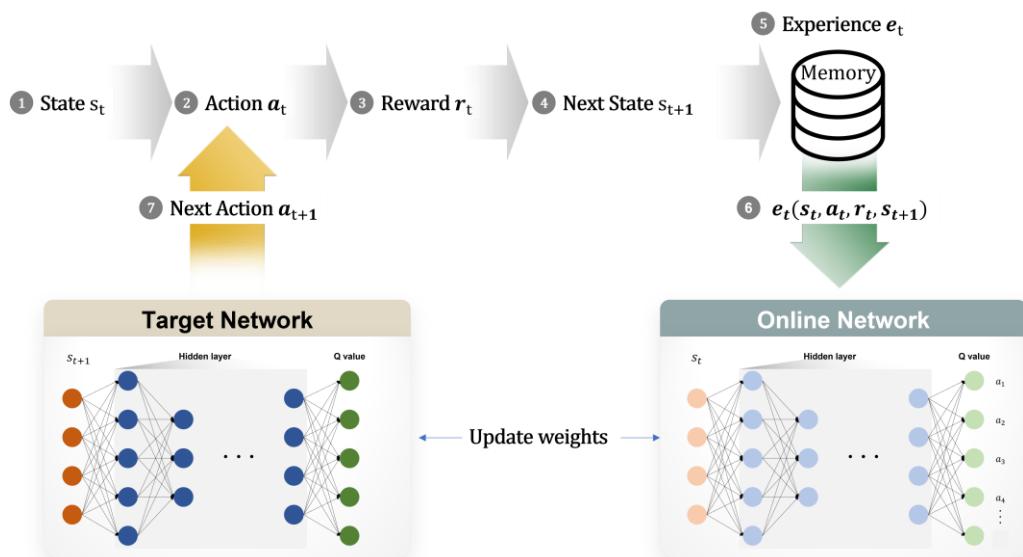


Figure 3. Schematic flows (states input and configured actions) for efficient computational rewards.

Building the neural network architecture that will serve as the online network in the DRL-SDNC will keep the action configuration or policy enforcement interactive and autonomous for the proposed UAV-MEC systems. Depending on the characteristics of the state representation and the temporal dynamics of the UAV-MEC networking environment, the weights are exchanged between online and target networks for iterative improvement. Experience gathering enables the agent to interact with the UAV-MEC networking environment and obtain the hidden patterns. The proposed agent acts as double Q-networks and updates the parameters of the online network using optimization methods like stochastic gradient descent and perform the iterative learning. The UAV-MEC environment states are gathered by the agent to gain new experiences, update the online network, and improve its offloading decision-making capabilities. SDNC monitors the performance of the agent during training and DRL-SDNC collaboratively adjusts hyperparameters or network architecture to enhance learning efficiency and achieve the long-term computational efficiency of the offloading tasks. Therefore, the proposed DRL-SDNC can learn to make intelligent decisions in a UAV-assisted networking environment, optimizing resource

efficiency based on the evaluated reward function and achieving efficient utilization of computing/communicating resources.

In the process, first, the proposed DRL agent receives input about the network state and flow statuses from the SDNC. Neural networks are used to estimate the value or policy function based on s_t . Next, the DRL-SDNC agent configures action a_t , which represents a flow scheduling decision, and communicates it to the SDNC. The SDNC applies the action by updating the flow rules through UPF. The updated UAV network state s_{t+1} is then fed back to the DRL-SDNC, allowing it to observe the outcome and update its neural network parameters. This iterative process continues to refine the DRL-SDNC agent's flow scheduling strategies to the optima.

4. Performance Evaluation

4.1. Simulation Setup

Table 2 presents the primary parameter configuration used in this system. In this system flows, there are five primary phases:

1. **Network topology and environment settings:** A simulated SDN environment is created with a network topology consisting of switches, controllers, and hosts. Hosts can be customized as UEs and MECs with specific requirements of resource settings. In OpenAI, the replication of network topology is carried out as an environment initialization (with randomness and reset every episode) following the state tuple information as listed in Equation (3). The state presentation of UAV network is added using OpenAI to complete the missing topology in SDN, and later collected and represented in a suitable format for the DRL-SDNC agent. The state representation serves as input to the agent's neural network, which constructs using TensorFlow.
2. **Task offloading:** A set of tasks with random requirements and capacities, such as application complexity, latency constraints, and energy/resource consumption/requirement are generated. Each UAV-MEC pair represents a potential offloading option for task execution, and the flow rule is synced to central SDNC.
3. **DRL agent:** An instance of the DRL *agent function* using OpenAI libraries with TensorFlow is initialized. The *environment function* samples the state, and *action function* alters the state information into new space after every iteration action is applied. **Reset function** is executed when the episode ends, or optima is reached. **Immediate reward** and **Q-value function** is co-located with the *agent function* to evaluate the state-action pair performance.
4. **Training and iterative learning:** The proposed agent is trained using a sampling state matrix generated from the UAV-MEC environment. The agent's online/target network parameters are updated through backpropagation and gradient descent to optimize its policy/value function. The hyperparameters and network configurations are iteratively adjusted to optimize the ordering flow for task offloading in the UAV-MEC environment.
5. **Evaluation:** The reward per episode can be captured within OpenAI-TensorFlow simulation, and network configuration of sending payload task sizes based on agent action identifies the efficient/inefficient computational resources which possibly leads to high drop ratios.

Table 2. Primary parameter configuration for DRL-based SDN backbone environment.

Purpose/Platform	Specification
Hosting infrastructure	Intel(R) Xeon(R) Silver 4280 CPU @ 2.10 GHz, 128 GB, NVIDIA Quadro RTX 4000 GPU
Number of UAVs, MECs, and BS	5
Task generation rate, constraints, bandwidth, resource scales, channel gain, power, UAV coordination/distances, and speed	Ratio scale (0 to 1) for agent sampling and adjusting based on state-action forward-backward calculation
Task complexity and sizes	Random set (intensive, normal, non-intensive) ranging (256 Kbits, 512 Kbits, 1024 Kbits)
Learning rate	0.001
Discount factor	0.95
Batch size	Random set (32, 64, 128, 256) by congestion states
Exploration	0.5
Number of episodes	500
Simulation times	5000 s
SDN-UE, control, and interfaces	Mininet

With the integration of DRL and SDN [34,35], this simulation setup enables the DRL agent to learn and adapt its decision-making process to schedule for computational resource efficiency in UAV-MEC task offloading, leading to improved energy efficiency, reduced overhead latency, and enhanced resource utilization.

4.2. Proposed and Reference Schemes

This sub-section presents the proposed and reference schemes applied in the experiments to illustrate the performance differences in terms of different congestion conditions, task complexities, and intensity level of the heavy tasks compared with the simulation setting and episode numbers of the learning iteration.

- **PDRL-SDNC-UAV** indicates the proposed DRL-based SDN backbone control utilizing double Q-network to approximate the Q-value function and collaboratively configures the SDN flow rules for efficient scheduling and resource placement. This approach intelligently train the function approximator to handle high-dimensional state space observation and representation. The experience replay, network training, and SDNC synchronization in UAV-MEC networks follow the expression in Section 3 to learn complex policies for emphasizing the critical weights on sub-reward r_{cr}^t , and handles different congestion states from multi-diversity complex application with high computation-intensive tasks.
- **T-SDNC-UAV** represents the traditional SDNC for UAV-MEC-assisted network environment with task offloading problem handling. This baseline involves a centralized controller that manages the network's resources and controls the behavior of the UAVs, which obeys the default SDN rules. The controller communicates with the UAVs and makes decisions based on network-wide information, such as topology, traffic conditions, and default offloading policies. T-SDNC-UAV approach focuses on network-level optimization, ensuring efficient routing, resource allocation, and QoS provisioning for UAV-assisted networks.
- **SRL-SDNC-UAV** indicates the single reinforcement learning-based SDNC approach, in which a single Q-learning is employed to control the UAV network. The agent learns through trial and error, optimizing its policy by iterative querying through Q-learning processing (e.g., using Q-table).

4.3. Results and Discussion

In this subsection, we present the results of the proposed *PDRL-SDNC-UAV* and reference schemes, namely *SRL-SDNC-UAV* and *T-SDNC-UAV*, in terms of rewards and task delivery/drop ratios, primarily based on the efficient/inefficient computational resource conditions. The simulation setup conducts on the same topology; only the controller and agent are different for the performance metrics.

Figure 4 demonstrates the total rewards R_{env}^t throughout each scheme's exploration and exploitation within 500 episodes. In the exploration phase, the proposed scheme, *PDRL-SDNC-UAV*, achieves a significantly higher R_{env}^t compared to other schemes, with a difference of 11 and 32 positive scores when compared to *SRL-SDNC-UAV* and *T-SDNC-UAV*, respectively. In the exploitation phase, deep (RL)-based approaches achieve a completely stable reward compared to traditional control, which is 60% difference. At the ending episode, the proposed scheme reaches 34 positive scores which is 11.76% and 47.05% higher efficiency points compared to *SRL-SDNC-UAV* and *T-SDNC-UAV*, respectively. Figure 5 illustrates the main scoring metrics, which is r_{cr}^t . The proposed scheme configured the emphasized weight ω for efficient computational resources, which achieves 50% of the total rewards for this particular sub-reward, compared to 30% and 25% from *SRL-SDNC-UAV* and *T-SDNC-UAV*, respectively. *PDRL-SDNC-UAV*, *SRL-SDNC-UAV* and *T-SDNC-UAV* reached 17, 9, and 4.5, respectively, at the ending episode, which shows that the proposed scheme highly outperforms both baseline approaches.

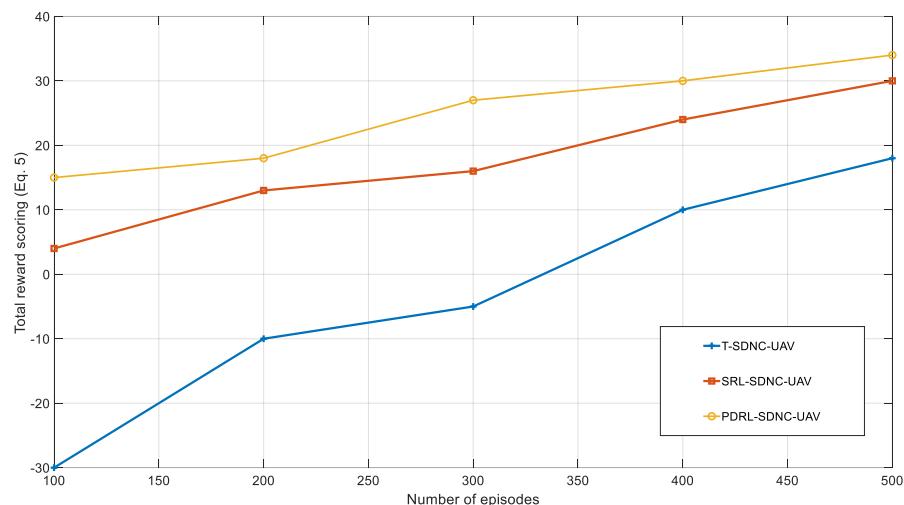


Figure 4. Performance metrics on total reward scoring.

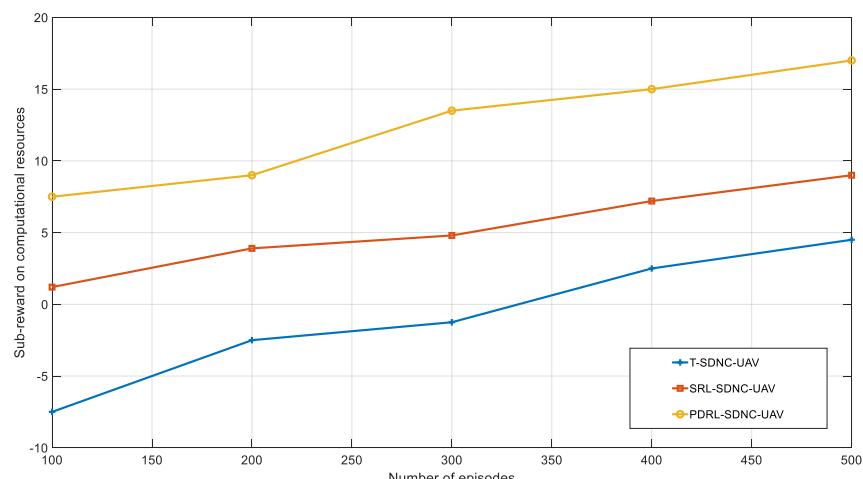


Figure 5. Performance metrics on sub-reward of computational resources.

The sub-rewards on energy r_e^t and latency r_d^t scoring points, presented in Figures 6 and 7, are primarily used to regulate the upper-bound tolerable delays and do not exceed the remaining energy (e_i^t , e_n^t), which leads the proposed scheme to slice only 25% for these two sub-rewards. For *T-SDNC-UAV*, the result on energy sub-reward is peaked due to default SDNC configuration following the energy limitation constraints. For result on latency sub-reward, *SRL-SDNC-UAV* achieved higher immediate reward than the proposed scheme due to the single processing compared to double network architecture; however, the efficiency of configured action remained lower than the proposed scheme. Regarding reward efficiency perspective, this proposed scheme handles dynamic adjustment offloading paths with efficient computing capacities, and the selection/placement actions/decisions on UAV-MEC pair node are highly adequate for improving the resource-constrained or computation-intensive tasks and also for adaptability in UAV-MEC-assisted network environments.

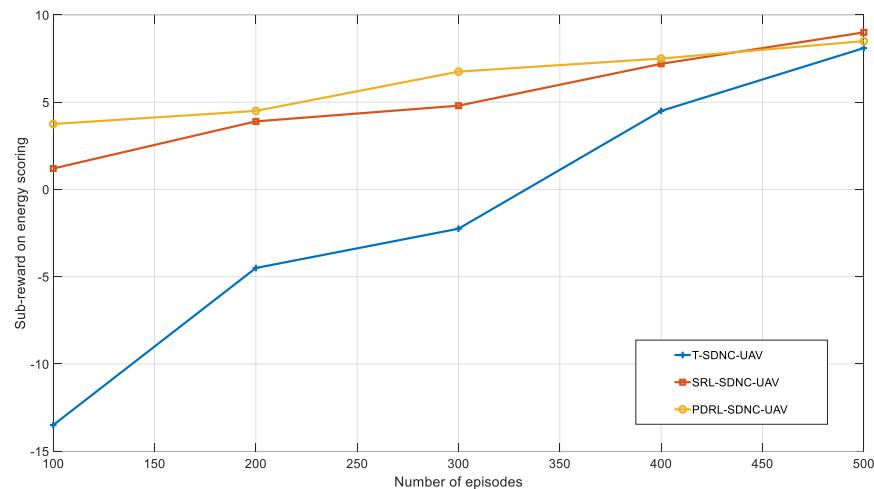


Figure 6. Performance metrics on sub-reward of energy scoring.

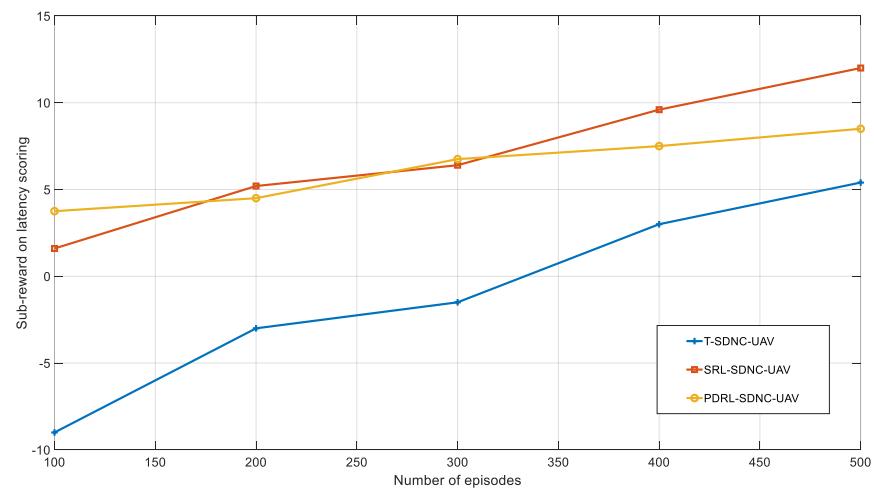


Figure 7. Performance metrics on sub-reward of latency scoring.

For task successful/failure ratios, the results are illustrated in Figures 8 and 9, where network states are consecutively configured by increasing congestion levels and application task complexity per 1000 simulation interval. We evaluate the agent flexibility by observing the result fluctuation differences compared to the diversity of heavy task and congestion states. The proposed scheme has the least fluctuation from 1000 to 5000 simulation times/conditions. *PDRL-SDNC-UAV* achieved the successful ratios (efficient offloading to adequate computational resources) from 99.91% to 99.98% (high to low intensity), which is (0.5%, and 0.03%) and (0.3% and 0.02%) better than *T-SDNC-UAV* and *SRL-SDNC-UAV*,

respectively. In high computation-intensive and task generation rates, the failure rate of baseline approaches reached 0.59% and 0.39%, respectively. The proposed scheme indicates the policy reliability in achieving long-term adequacy, especially for multiple taxonomy-diversified IoT tasks. The resource-aware algorithm ensures the reliability and transferability of scalable UAV-MEC management systems by still efficiently considering the multi-aspect weighted sum reward on energy and delays.

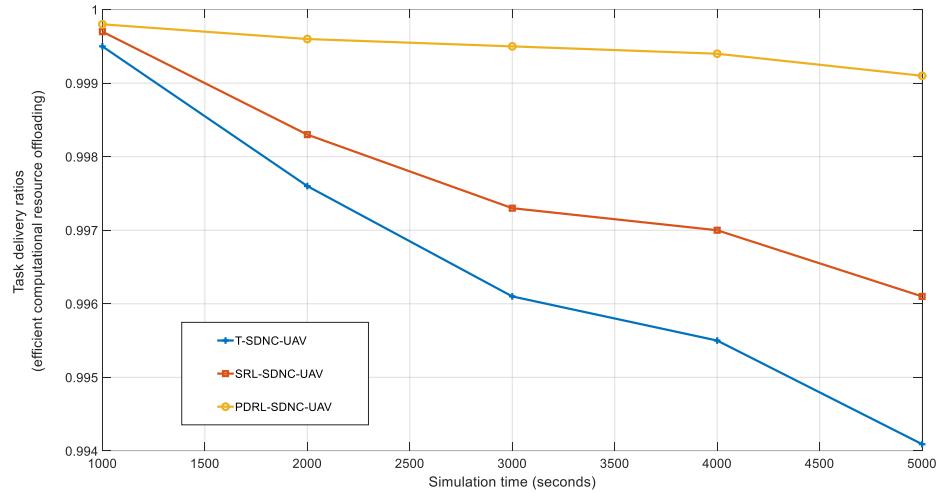


Figure 8. Performance metrics on task delivery ratios (successful rates from efficient computational resource control).

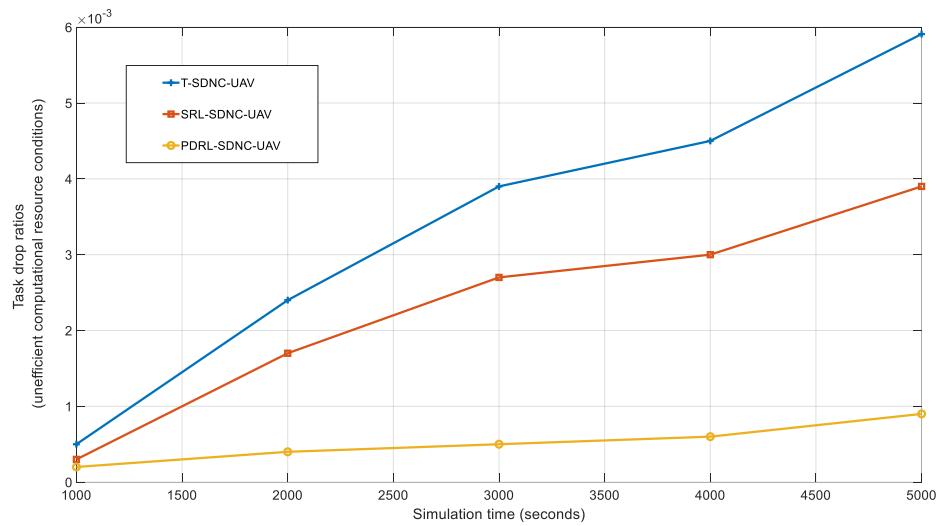


Figure 9. Performance metrics on task drop ratios (failure rates from inefficient selection/placement).

5. Conclusions and Future Work

This paper proposed a novel approach based on DRL for collaborating with SDN backbone control in UAV-assisted networks, termed PDRL-SDNC-UAV. The main objective was to develop an intelligent agent with primary sub-reward function on computational resource efficiency, and partial sub-rewards to meet the requirement on energy and delays. This paper presented the system architecture and interfaces, including the task processing in SDN-enabled 5G network systems. The components of DRL modeling are given, including the state observation, configurable actions in SDN policies and offloading decision, and reward objectives. Experimental simulations using OpenAI-TensorFlow and SDN emulator were conducted to evaluate the performance of the proposed approach compared to reference approaches in UAV-assisted networks. The evaluation considered reward measurements in terms of delay, energy, and resource. The task successful/failure is described

to point out the stability of policy effectiveness over various network state conditions. *PDRL-SDNC-UAV* scheme outperformed the *SRL-SDNC-UAV* and *T-SDNC-UAV* schemes and achieved significantly higher rewards and more efficient resource utilization, indicating its effectiveness in optimizing UAV-assisted networking environments. Furthermore, the proposed scheme showed high reliability, even under varying levels of congestion and task complexity for scalable and long-term UAV-MEC management systems.

In future studies, an integrated softwareization on OpenAI-agent and SDN policy will be conducted and the extensive formulation on actor-critic will be used to advance this work for heterogeneous multi-UAV multi-MEC aspects. The state enhancement and interaction flow of each functionality will be extended for multi-objective awareness and demonstrated on improved DRL-based edge resource management.

Author Contributions: Conceptualization, I.S., P.T. and S.K. (Seokhoon Kim); methodology, I.S. and P.T.; software, P.T. and S.K. (Seungwoo Kang); validation, S.K. (Seungwoo Kang), I.S. and S.R.; formal analysis, S.K. (Seungwoo Kang), I.S. and S.R.; investigation, S.K. (Seokhoon Kim); resources, S.K. (Seokhoon Kim); data curation, P.T.; writing—original draft preparation, I.S. and P.T.; writing—review and editing, I.S. and P.T.; visualization, I.S.; supervision, S.K. (Seokhoon Kim); project administration, S.K. (Seokhoon Kim); funding acquisition, S.K. (Seokhoon Kim). All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (No. RS-2022-00167197, Development of Intelligent 5G/6G Infrastructure Technology for The Smart City), in part by BK21 FOUR (Fostering Outstanding Universities for Research) under Grant 5199990914048, in part by the National Research Foundation of Korea (NRF), Ministry of Education, through Basic Science Research Program under Grant NRF-2020R1I1A3066543, and in part by the Soonchunhyang University Research Fund.

Data Availability Statement: Not applicable.

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

References

1. Mao, Y.; You, C.; Zhang, J.; Huang, K.; Letaief, K.B. A Survey on Mobile Edge Computing: The Communication Perspective. *IEEE Commun. Surv. Tutor.* **2017**, *19*, 2322–2358. [[CrossRef](#)]
2. Vhora, F.; Gandhi, J.C. A Comprehensive Survey on Mobile Edge Computing: Challenges, Tools, Applications. In Proceedings of the 2020 Fourth International Conference on Computing Methodologies and Communication (ICCMC), Erode, India, 11–13 March 2020; pp. 49–55.
3. Taleb, T.; Samdanis, K.; Mada, B.; Flinck, H.; Dutta, S.; Sabella, D. On Multi-Access Edge Computing: A Survey of the Emerging 5G Network Edge Cloud Architecture and Orchestration. *IEEE Commun. Surv. Tutor.* **2017**, *19*, 1657–1681. [[CrossRef](#)]
4. Duan, Q.; Wang, S.; Ansari, N. Convergence of Networking and Cloud/Edge Computing: Status, Challenges, and Opportunities. *IEEE Netw.* **2020**, *34*, 148–155. [[CrossRef](#)]
5. Sha, D.; Zhao, R. DRL-Based Task Offloading and Resource Allocation in Multi-UAV-MEC Network with SDN. In Proceedings of the 2021 IEEE/CIC International Conference on Communications in China (ICCC), Xiamen, China, 28–30 July 2021.
6. Do, Q.V.; Pham, Q.-V.; Hwang, W.-J. Deep Reinforcement Learning for Energy-Efficient Federated Learning in UAV-Enabled Wireless Powered Networks. *IEEE Commun. Lett.* **2022**, *26*, 99–103. [[CrossRef](#)]
7. Geraci, G.; Garcia-Rodriguez, A.; Azari, M.M.; Lozano, A.; Mezzavilla, M.; Chatzinotas, S.; Chen, Y.; Rangan, S.; Renzo, M.D. What Will the Future of UAV Cellular Communications Be? A Flight from 5G to 6G. *IEEE Commun. Surv. Tutor.* **2022**, *24*, 1304–1335. [[CrossRef](#)]
8. Li, B.; Fei, Z.; Zhang, Y. UAV Communications for 5G and Beyond: Recent Advances and Future Trends. *IEEE Internet Things J.* **2019**, *6*, 2241–2263. [[CrossRef](#)]
9. Kim, J.; Lee, J.; Yang, E.; Kang, S. Technology Forecasting from the Perspective of Integration of Technologies: Drone Technology. *KSII Trans. Internet Inf. Syst.* **2023**, *17*, 31–50.
10. Abdulghaffar, A.; Mahmoud, A.; Abu-Amara, M.; Sheltami, T. Modeling and Evaluation of Software Defined Networking Based 5G Core Network Architecture. *IEEE Access* **2021**, *9*, 10179–10198. [[CrossRef](#)]
11. Kiran, N.; Liu, X.; Wang, S.; Yin, C. VNF Placement and Resource Allocation in SDN/NFV-Enabled MEC Networks. In Proceedings of the 2020 IEEE Wireless Communications and Networking Conference Workshops (WCNCW), Seoul, Republic of Korea, 6–9 April 2020; pp. 1–6.

12. Hu, Y.; Zhu, L.; Zhang, J.; Cai, Z.; Han, J. Migration and Energy Aware Network Traffic Prediction Method Based on LSTM in NFV Environment. *KSII Trans. Internet Inf. Syst.* **2023**, *17*, 896–915.
13. Qiao, Q. Routing Optimization Algorithm for Logistics Virtual Monitoring Based on VNF Dynamic Deployment. *KSII Trans. Internet Inf. Syst.* **2022**, *16*, 1708–1734.
14. Tran, G.K.; Ozasa, M.; Nakazato, J. NFV/SDN as an Enabler for Dynamic Placement Method of mmWave Embedded UAV Access Base Stations. *Network* **2022**, *2*, 479–499. [[CrossRef](#)]
15. Zheng, J.; Tian, C.; Dai, H.; Ma, Q.; Zhang, W.; Chen, G.; Zhang, G. Optimizing NFV Chain Deployment in Software-Defined Cellular Core. *IEEE J. Sel. Areas Commun.* **2020**, *38*, 248–262. [[CrossRef](#)]
16. Tam, P.; Song, I.; Kang, S.; Ros, S.; Kim, S. Graph Neural Networks for Intelligent Modelling in Network Management and Orchestration: A Survey on Communications. *Electronics* **2022**, *11*, 3371. [[CrossRef](#)]
17. Ros, S.; Eang, C.; Tam, P.; Kim, S. ML/SDN-Based MEC Resource Management for QoS Assurances. In *Advances in Computer Science and Ubiquitous Computing*; Springer: Singapore, 2023; Volume 1028, pp. 591–597.
18. Ouamri, M.; Alkanhel, R.; Singh, D.; El-kenaway, E.S.M.; Ghoneim, S.S. Double Deep Q-Network Method for Energy Efficiency and Throughput in a UAV-Assisted Terrestrial Network. *Comput. Syst. Eng.* **2023**, *46*, 73–92. [[CrossRef](#)]
19. Ouamri, M.; Alkanhel, R.; Gueguen, C.; Alohal, M.; Ghoneim, S.S. Modeling and Analysis of UAV-Assisted Mobile Network with Imperfect Beam Alignment. *Comput. Mater. Contin.* **2022**, *74*, 453–467. [[CrossRef](#)]
20. Liu, Q.; Cheng, L.; Jia, A.L.; Liu, C. Deep Reinforcement Learning for Communication Flow Control in Wireless Mesh Networks. *IEEE Netw.* **2021**, *35*, 112–119. [[CrossRef](#)]
21. Tian, A.; Feng, B.; Zhou, H.; Huang, Y.; Sood, K.; Yu, S.; Zhang, H. Efficient Federated DRL-Based Cooperative Caching for Mobile Edge Networks. *IEEE Trans. Netw. Serv. Manag.* **2022**, *20*, 246–260. [[CrossRef](#)]
22. Tam, P.; Corrado, R.; Eang, C.; Kim, S. Applicability of Deep Reinforcement Learning for Efficient Federated Learning in Massive IoT Communications. *Appl. Sci.* **2023**, *13*, 3083. [[CrossRef](#)]
23. Yang, Z.; Pan, C.; Wang, K.; Shikh-Bahaei, M. Energy Efficient Resource Allocation in UAV-Enabled Mobile Edge Computing Networks. *IEEE Trans. Commun.* **2019**, *18*, 4576–4589. [[CrossRef](#)]
24. He, Y.; Zhai, D.; Huang, F.; Wang, D.; Tang, X.; Zhang, R. Joint Task Offloading, Resource Allocation, and Security Assurance for Mobile Edge Computing-Enabled UAV-Assisted VANETs. *Remote Sens.* **2021**, *13*, 1547. [[CrossRef](#)]
25. Tan, T.; Zhao, M.; Zeng, Z. Joint Offloading and Resource Allocation Based on UAV-Assisted Mobile Edge Computing. *ACM Trans. Sens. Netw. (TOSN)* **2022**, *18*, 1–21. [[CrossRef](#)]
26. Munaye, Y.Y.; Juang, R.-T.; Lin, H.-P.; Tarekgn, G.B.; Lin, D.-B. Deep Reinforcement Learning Based Resource Management in UAV-Assisted IoT Networks. *Appl. Sci.* **2021**, *11*, 2163. [[CrossRef](#)]
27. Yu, F.; Yang, D.; Wu, F.; Wang, Y.; He, H. Resource Optimization for UAV-Assisted Mobile Edge Computing System Based on Deep Reinforcement Learning. *Phys. Commun.* **2023**, *59*, 102107. [[CrossRef](#)]
28. Ren, Y.; Guo, A.; Song, C. Multi-Slice Joint Task Offloading and Resource Allocation Scheme for Massive MIMO Enabled Network. *KSII Trans. Internet Inf. Syst.* **2023**, *17*, 794–815.
29. Song, I.; Kang, S.; Tam, P.; Kim, S. Federated Logistic Regression for Reliable Prediction Models in Privacy-Preserving Healthcare Networks. In Proceedings of the 2022 6th International Conference on Interdisciplinary Research on Computer Science, Psychology, and Education (ICICPE' 2022), Pattaya, Thailand, 31 October 2022.
30. Zhang, P.; Su, Y.; Li, B.; Liu, L.; Wang, C.; Zhang, W.; Tan, L. Deep Reinforcement Learning Based Computation Offloading in UAV-Assisted Edge Computing. *Drones* **2023**, *7*, 213. [[CrossRef](#)]
31. Lin, C.; Han, G.; Shah, S.B.H.; Zou, Y.; Gou, L. Integrating Mobile Edge Computing into Unmanned Aerial Vehicle Networks: An SDN-Enabled Architecture. *IEEE Internet Things Mag.* **2021**, *4*, 18–23. [[CrossRef](#)]
32. Sellami, B.; Hakiri, A.; Ben Yahia, S.; Berthou, P. Deep Reinforcement Learning for Energy-Efficient Task Scheduling in SDN-Based IoT Network. In Proceedings of the 2020 IEEE 19th International Symposium on Network Computing and Applications (NCA), Cambridge, MA, USA, 24–27 November 2020; pp. 1–4.
33. Zhang, L.; Lu, Y.; Zhang, D.; Cheng, H.; Dong, P. DSOQR: Deep Reinforcement Learning for Online QoS Routing in SDN-Based Networks. *Secur. Commun. Netw.* **2022**, *2022*, 4457645. [[CrossRef](#)]
34. Tam, P.; Song, I.; Kang, S.; Kim, S. Privacy-Aware Intelligent Healthcare Services with Federated Learning Architecture and Reinforcement Learning Agent. In *Advances in Computer Science and Ubiquitous Computing*; Springer: Singapore, 2023; Volume 1028, pp. 583–590.
35. Ros, S.; Tam, P.; Kim, S. Modified Deep Reinforcement Learning Agent for Dynamic Resource Placement in IoT Network Slicing. *J. Internet Serv. Appl.* **2022**, *23*, 17–23.

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.