aerospace

MDPI

Article

AI-Enabled Interference Mitigation for Autonomous Aerial Vehicles in Urban 5G Networks †

Anirudh Warrier *,‡, Saba Al-Rubaye ‡, Gokhan Inalhan ‡ and Antonios Tsourdos ‡

School of Aerospace, Transport and Manufacturing (SATM), Cranfield Univeristy, College Road, Bedford MK43 0AL, UK

* Correspondence: anirudh.warrier@cranfield.ac.uk

† This paper is an extended version of our paper published in 2022 IEEE/AIAA 41st Digital Avionics Systems (DASC).

‡ These authors contributed equally to this work.

Abstract: Integrating autonomous unmanned aerial vehicles (UAVs) with fifth-generation (5G) networks presents a significant challenge due to network interference. UAVs' high altitude and propagation conditions increase vulnerability to interference from neighbouring 5G base stations (gNBs) in the downlink direction. This paper proposes a novel deep reinforcement learning algorithm, powered by AI, to address interference through power control. By formulating and solving a signal-to-interference-and-noise ratio (SINR) optimization problem using the deep Q-learning (DQL) algorithm, interference is effectively mitigated, and link performance is improved. Performance comparison with existing interference mitigation schemes, such as fixed power allocation (FPA), tabular Q-learning, particle swarm optimization, and game theory demonstrates the superiority of the DQL algorithm, where it outperforms the next best method by 41.66% and converges to an optimal solution faster. It is also observed that, at higher speeds, the UAV sees only a 10.52% decrease in performance, which means the algorithm is able to perform effectively at high speeds. The proposed solution effectively integrates UAVs with 5G networks, mitigates interference, and enhances link performance, offering a significant advancement in this field.

Keywords: autonomous vehicles; unmanned aerial vehicles (UAVs); fifth-generation (5G); interference mitigation; artificial intelligence; deep Q-learning

Citation: Warrier, A.; Al-Rubaye, S.; Inalhan, G.; Tsourdos, A. AI-Enabled Interference Mitigation for Autonomous Aerial Vehicles in Urban 5G Networks. *Aerospace* **2023**, *10*, 884. https://doi.org/10.3390/aerospace10100884

Academic Editors: Hailong Huang and Guohao Zhang

Received: 14 June 2023
Revised: 5 October 2023
Accepted: 9 October 2023
Published: 13 October 2023

1. Introduction

Over the past few years, the adoption of unmanned aerial vehicles (UAVs) or drones in commercial and civilian applications has increased significantly due to advances in manufacturing processes, resulting in their economic feasibility. This increase in usage has opened up an extensive range of potential applications, including photography, aerial inspection, disaster relief, traffic control, precision agriculture, delivery systems, and communications. Consequently, regulatory bodies and authorities have developed frameworks to ensure the safe and effective use of UAVs. For example, the Federal Aviation Administration (FAA) in the United States has issued operational regulations for small unmanned aircraft systems (UASs) weighing under 25 kg for civilian applications. Additionally, the FAA has launched the "Drone Integration Pilot Program" to facilitate the exploration of expanded UAV applications. This program includes rules for night flights, applications requiring UAVs to fly over people, and beyond visual line of sight (BVLoS) operations [1].

The issue of communication with cellular-connected UAVs has gained significant attention in recent times [2]. Existing cellular networks have not been specifically optimised to support UAVs and are primarily focused on providing high-quality service to terrestrial users. Given the stringent regulatory requirements, the need for extensive research and standardization efforts to ensure the reliable operation of UAVs in a range of deployment

scenarios is evident. These factors highlight the importance of optimizing current cellular networks to facilitate seamless communication with UAVs.

Novel guidelines and programs like the FAA's "Drone Integration Pilot Program" are expected to accelerate the growth of the UAV industry globally. Thus, UAVs are seen as an interesting opportunity for businesses in the next decade.

One of the prominent technologies to enable UAV communications is fifth-generation (5G) cellular technology. Indeed, 5G technology will ultimately replace fourth-generation (4G) networks as the new standard cellular networks. Currently, it provides connectivity to most cell phones, and 5G is beneficial because it is expected to provide higher download speeds and greater bandwidth. The future infrastructure of 5G networks can therefore link seamlessly and ubiquitously with everything. For instance, one attractive and growing technology is the Internet of Things (IoT), which has generated a tremendous increase in data traffic, thus leading to the requirement of advanced networks to cope with reliability and traffic requirements.

The realization of 5G and beyond 5G (B5G) wireless networks is crucial to meet the requirements of connected UAVs. These networks offer low latency, high speed, massive capacity, and edge computing, which can enable UAVs to operate virtually in real time. This translates to fewer incidents of loss of control and mid-air collisions, making 5G networks a critical component for the safe and efficient operation of UAVs. Therefore, there is an urgent need to accelerate the development and deployment of 5G networks to support the growth of the UAV industry [3].

Although 5G networks can provide high data rates and low latency, they may not be fully optimised for UAVs and may face challenges such as signal blockage and limited coverage. Therefore, we may need to consider using sixth-generation (6G) networks, which are expected to provide advanced features such as intelligent networking, enhanced sensing and perception, and autonomous decision making to resolve current challenges in communication with UAVs. However, it should be noted that the scope of our project does not include the development of 6G networks [4,5].

The utilization of UAVs in communication confers various promising advantages, including the ability for ad-hoc deployment, adaptable network reconfiguration, and a higher probability of achieving line of sight (LOS) links. UAVs that function as 5G base stations, referred to as Next Generation Node Bs (gNBs), represent one of the most remarkable examples and are anticipated to perform a critical role in forthcoming generations of wireless networks, as stated by [6]. The benefits provided by UAV-gNBs, like swift deployment, mobility, increased likelihood of unimpeded propagation, and flexibility, have gained extensive interest and are being widely investigated.

Figure 1 demonstrates several use cases for 5G, including (i) serving as a backup in areas where a gNB malfunction occurs; (ii) providing coverage in rural areas with limited or no gNB infrastructure; and (iii) enhancing quality of service (QoS) in areas with unexpected surges in demand, such as densely populated sports events, by establishing a network connected to the 5G core network via a backhaul node. Further details on the software-based architecture (SBA) of 5G will be presented in the following sections.

At present, communication between UAVs and the ground is restricted to visual line of sight (VLoS) range, which limits reliability, coverage, security, and performance. To overcome these limitations, it is crucial to connect drones to cellular networks to enable BVLoS capabilities. While integrating drones with 5G offers several advantages, it also poses certain challenges [7–11].

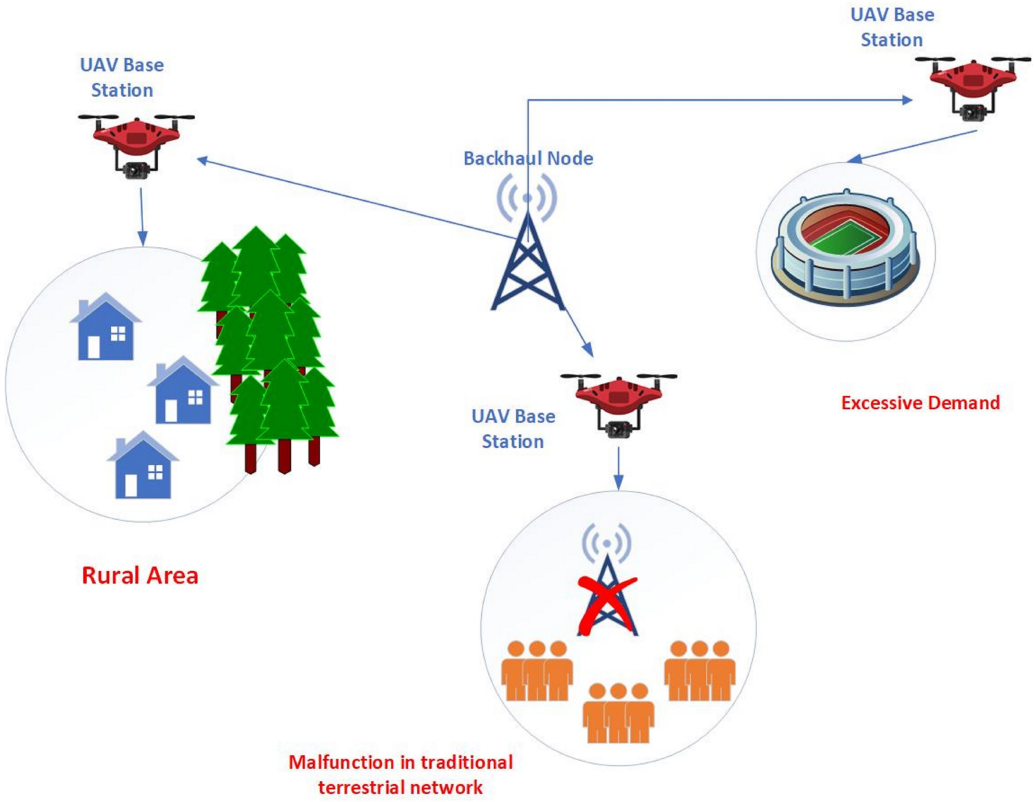

Figure 1. 5G-connected UAVs.

Contributions

The purpose of this paper is to develop a technique for performing power control to mitigate interference in UAVs operating within 5G networks. To achieve this goal, a novel method of power control is proposed, deviating from conventional approaches. Specifically, this method regulates transmit power not only at the serving gNB but also controls transmit powers of interfering gNBs from a central location. This approach introduces a race condition where the serving gNB for one UAV can act as an interfering gNB for another user. To address this challenge, deep reinforcement learning is employed due to its proven effectiveness in resolving similar problems.

The deep Q-learning (DQL) algorithm is considered a promising solution to the interference problem as it eliminates the need for reporting channel state information (CSI), thereby reducing overhead and resulting in a low-complexity system design. Under the DQL approach, UAVs report only their received signal-to-interference-and-noise ratio (SINR) and coordinates, while the gNB, supported by a cloud-based architecture, performs power control to mitigate interference. This approach involves both associated and interfering gNBs issuing power control commands, diverging from industry standards where only the associated gNB is involved. Building upon our previous work [12], this paper presents the following significant contributions:

- Develops a power control optimization problem for the downlink direction to maximise the SINR experienced by the UAV.
- Develops a framework based on deep reinforcement learning (DRL) that enables the concurrent execution of multiple actions and dynamically adjusts power levels to achieve optimal SINR by utilizing data from a given dataset.

Thus, the paper presents a comprehensive framework based on DRL to optimise the SINR experienced by UAVs through dynamic adjustment of power levels using data from a provided dataset. The paper follows a systematic approach, beginning with a literature review and identification of research gaps in Section 2. This is followed by

a discussion of the challenges posed by current interference mitigation techniques in Section 3. Section 4 provides a detailed analysis of 5G and UAV technology and their functionalities. In Section 5, the paper introduces DRL, its components, neural network architecture, and policy selection methods. Section 6 presents a comparison of existing interference mitigation techniques with the proposed solution and describes the proposed algorithm. The simulation environment, setup, and results obtained are discussed in Section 7. The paper concludes with a discussion on the impact assessment of the results in Section 8.

2. Background and Related Works

The utilization of UAVs for meeting the growing communication needs has been widely advocated. Despite this, operation of UAVs at high altitude would enable LoS links to be significantly pervasive within the viable communication channels; as a result, a UAV has a high likelihood of communicating with both associated gNBs and interfering gNBs simultaneously. In the case of ground users, who are served by the primary lobe of the gNB antenna, an escalation in pathloss typically correlates with a reduction in interfering signals. However, when UAVs are served by the side lobes of the gNB antenna, the relationship between increasing pathloss and interference becomes less straightforward. This complexity arises because side lobes have the potential to introduce unintended coverage areas and additional propagation paths, thereby giving rise to interference with other systems or gNBs. Consequently, even when the main lobe signal strength is weakened due to pathloss, these side lobes can still generate interference. This scenario increases the likelihood of interference as compared to ground-based users, posing a significant challenge for interference management in cellular networks that include UAVs. Several techniques have been proposed in extant literature to mitigate interference in terrestrial networks, including inter-cell interference coordination (ICIC) [13,14] and coordinated multi-point (CoMP) transmission [15,16]. However, these techniques prove inadequate in handling new interference challenges arising from UAVs and the emergence of LoS-dominated air–ground links at high altitudes.

ICIC is a radio resource management technique used to enhance spectral efficiency. This is achieved by imposing constraints on the management block to improve favourable channel conditions for a subset of users significantly affected by interference. The coordination of resources may be achieved through fixed, adaptive, or real-time approaches, supported by additional inter-cell signalling, with the frequency adjusted based on specific requirements. A cooperative interference cancellation technique is proposed in [17] to minimise the sum rate to available gNBs and eliminate co-channel interference at each occupied gNB for multi-beam UAV uplink communication. Although similar techniques have been extensively studied in terrestrial cellular networks [18], they are not effective in mitigating strong UAV interference, leading to limited frequency reuse in the network and lower spectral efficiency for both UAVs and ground users [19]. In [20], the interference in directional UAV networks is characterised using stochastic geometry, considering UAVs equipped with directional antennas and situated in a 3D environment. This work provides a comprehensive analysis of the interference in directional UAV networks, but no strategies are presented for mitigating it.

The approach of employing a path planning or trajectory design to avoid or minimise interference is a widely adopted strategy for interference management. In the paper [21], the authors consider a relay-assisted UAV network overlaid with an existing network and propose a joint power and 3D trajectory design approach to reposition the UAVs in 3D space to circumvent interference. They propose a joint optimisation solution based on spectral graph theory and convex optimisation to address the problem of 3D trajectory design and power allocation. The simulation results demonstrate that the proposed algorithm improves the maximum flow and reduces undesirable interference to the coexisting network [22].

The authors of [23] explore the joint trajectory and power control (TPC) design problem for multi-user UAV-enabled interference channels (UAV-IC). As the TPC problem is NP-hard and involves a significant number of optimisation variables, the authors propose effective sub-optimal algorithms. Although a multi-UAV setup is considered in both [22] and [23], the trajectory optimisation approach may not be optimal in situations where the path or trajectory of the UAV is unknown beforehand or changes during the mission.

In [24], an alternative solution to mitigate UAV–ground interference is proposed by utilizing intelligent reflective surfaces (IRSs) near gNBs for both downlink and uplink communications. The authors suggest an optimal passive beamforming design using IRSs for the downlink to minimise terrestrial interference with UAVs, and a hybrid linear and nonlinear interference coordination (IC) scheme in the uplink to handle the strong interference from UAVs. The authors further investigate the optimal performance of this scheme. However, it should be noted that IRSs are unsuitable for digital applications as they are designed based on the concept of analog beamforming. In [25], an interference-aware energy-efficient scheme is proposed to allow cellular-connected aerial users to reduce the interference they impose on terrestrial users while maximizing their energy efficiency and spectral efficiency. The optimization problem is formulated considering the key performance indicator (KPI) requirements for both aerial and terrestrial users, including energy efficiency, spectral efficiency, and interference. The problem is solved through the use of a deep Q-learning approach. However, it is critical to note the energy efficiency problem cannot be universally solved as it is dependent on various factors, such as mission design, UAV capabilities, and UAV size.

In recent years, artificial intelligence (AI) has emerged as a critical technology for 5G-and-beyond wireless networks that include UAVs, as demonstrated in studies like [26]. This is primarily due to AI's capacity to handle complex problems and large amounts of data in system design and optimisation, making it a crucial tool for creating highly dynamic UAV communication networks. Conventional offline and model-driven trajectory design methods are limited in practical scenarios with variable traffic and open operating environments due to the requirement for accurate communication models and complete understanding of system parameters. However, with the use of deep reinforcement learning, UAVs can predict future network states in real time and adjust communication resource allocation and UAV trajectories accordingly. Additionally, AI-embedded UAVs can play a significant role in edge computing applications, where multiple UAVs can collaborate as aerial edge servers or edge devices for efficient data and computation offloading.

Several research studies have investigated the application of DRL in the field of communications in recent years, as evidenced by publications such as [8,27–29]. In particular, ref. [29] employs DRL for power regulation in millimeter-wave (mm-wave) transmission as an alternative to beamforming for enhancing non-line-of-sight (NLoS) performance. The authors employ DRL to tackle the power allocation problem with the objective of maximizing the aggregate rate of the user equipment (UE) while meeting quality and transmission power constraints. The proposed approach involves estimating the Q-function of the DRL problem using a convolutional neural network. In another work, ref. [30] introduces a policy framework that employs DQL to maximise the number of transmissions in a multi-channel access network. In [31], the authors propose a joint optimization approach that employs a Q-learning algorithm to optimise power control, interference coordination, and communication performance for end-users in a 5G wireless network, enhancing the downlink SINR in a multi-access Orthogonal Frequency Division Multiplexing (OFDM) network from a gNB with multiple antennas to UEs with a single antenna. Finally, ref. [32] uses DQL to develop a policy to maximise transmissions in a dynamic correlated multi-channel access scenario. It is important to note, however, that these studies only address interference in the context of mobile devices. Neural networks were used in [28] to forecast mm-wave beams with low training overhead utilising received signals from nearby gNBs. Ref. [31] analysed voice signals for a multiple-access network consisting of many gNBs. The mid-band 5G frequency was only discussed in the [33] framework in

a single gNB scenario. Deep neural networks, which need channel knowledge to make decisions, were used in [33] to perform joint beamforming and interference cooperation at mm-wave. The effectiveness of deep neural networks for beamforming without using reinforcement learning was examined in [34]. These works have deployed machine learning algorithms to implement beamforming to solve the interference problem. Although beamforming improves the SINR of the system and resists interference, it is very complicated and expensive in terms of the hardware to be deployed. Advanced techniques such as double deep Q-learning were discussed in [35]. The main difference between DQL and double deep Q-learning (DDQL) is that DDQL uses two separate deep neural networks to estimate the Q-values, whereas DQL uses only one. The idea behind DDQL is to reduce the overestimation of Q-values that can occur in DQL. However, in some cases, such as when dealing with interference in 5G networks, overestimation may not be a problem and may even be desirable. In the case of interference in 5G networks, the goal is to minimise the interference between different channels and users. This can be achieved by optimizing the transmission power and frequency allocation of different users. In this scenario, DQL may be more suitable than DDQL because it allows for more exploration of the state–action space, which can lead to better performance in complex environments. Additionally, DQL has been shown to be more computationally efficient than DDQL, which can be a critical factor when dealing with large-scale 5G networks with many users and channels.

The literature review is summarised in Table 1 and the outcomes are discussed in the following section.

Table 1. Summary of literature.

	Optimisation Problem	**Algorithm**	**Gap**
[29]	Power Control for Downlink SINR	Deep Reinforcement Learning	It only considers the interference problem with terrestrial UEs and does not consider UAVs.
[30]	Power control for uplink power and sum rate	deep neural networks	It only considers the interference problem with terrestrial UEs and does not consider UAVs.
[36]	beamforming and power control for uplink sum-rate	convex optimisation	To ensure the optimal performance of the power domain concept at the receiver, it is imperative that the channel gain difference between users is appropriate. This requirement limits the effective number of user pairs that can be served by clusters.
[37]	beamforming for SINR	Q-learning	Q-learning becomes inefficient when the state–action space becomes excessively large
[38]	SINR	Mean-Field Game Theory	When the number of users in game theory increases, it is challenging to implement
[39]	Power control for SINR and spectral efficiency	CNN	It deals with energy harvesting networks, which is similar to a UAV but still does not take into account parameters like altitude
Proposed	Power control for SINR	Deep Q-learning	The proposed algorithm solves the problem of interference mitigation for UAVs owing to the higher altitude and favourable LoS characteristics

Outcome

UAVs are frequently touted as a promising solution for a wide range of applications, and, as such, the future 5G networks are expected to be able to support UAVs flying at higher altitudes and speeds with minimal interference.

Within the existing literature, many works discuss interference mitigation techniques; however, they also have certain drawbacks and unsuitability to the problem at hand. Methods such as ICIC, as discussed in [17–20], are effective for reducing interference via coordination; however, in dynamic and unpredictable environments, such as UAVs in 5G networks, they would be ineffective. In addition, they also generate high overhead, thus

increasing the complexity. Another method employed within the literature is path planning for UAVs, as discussed in [21,22]. This optimises the trajectory of the UAV so it can take the path with the least interference possible. One clear disadvantage of this would relate to not knowing the path of the UAV beforehand. In addition, dynamic routes could also render this method ineffective. Even though current AI/ML approaches such as [8,27–30] have been effective at mitigating interference, these methods have been deployed for UAVs in general or for users in 5G networks but not both, which is the unique problem that the proposed method in this paper solves. In Table 1, the literature is summarised and shows that the current solutions for mitigating interference are not sufficient. To address this issue, this paper proposes an interference mitigation approach building upon the work presented in [12].

3. Interference Mitigation Challenges

When comparing UAVs to traditional ground UEs, it is important to note that UAVs typically operate at higher altitudes. This presents a unique challenge for gNBs as they must provide 3D communication coverage instead of the usual 2D coverage for ground UEs. While conventional gNBs are angled downward to serve ground users and minimise inter-cell interference, UAVs must be integrated into this 3D communication network. This introduces new challenges that must be addressed in order to achieve effective and seamless communications. One of the most critical challenges in UAV communication within a cellular network is interference management. Interference in aerial users is exacerbated by LoS-dominated UAV-gNB channels, which is largely due to the high altitude of UAVs. As a result, this phenomenon represents a significant obstacle to the proper functioning of UAVs in the network. In summary, the integration of UAVs into cellular networks requires careful consideration of 3D coverage, interference management, and other challenges unique to aerial communication.

Figure 2 illustrates aerial interference in a 5G network. Specifically, it highlights the impact of height on interference for two UAVs, namely UAV1 and UAV2, which fly at altitudes h_1 and h_2, respectively. UAV1 experiences severe interference from a neighbouring gNB, whereas UAV2, flying at a greater altitude ($h_2 > h_1$), faces interference from three neighbouring gNBs due to the presence of favorable LoS links. This observation suggests that higher altitude leads to interference from all neighbouring gNBs due to the existence of favorable LoS links. During downlink communication, each UAV experiences interference from multiple neighbouring gNBs that are not associated with it due to strong LoS channels, resulting in poor downlink performance. In uplink communication, the UAV creates significant interference for several adjacent non-associated gNBs, thereby causing a new gNB interference problem. Although resource block (RB) allocation is a common approach for addressing this issue, it is ineffective for severe air–ground interference. This is because of the limited number of RBs available to UAVs and the high frequency reuse for terrestrial UEs in cellular networks. Thus, developing an interference mitigation technique that accounts for the unique channel and interference characteristics of cellular-connected UAVs is critical. Although current terrestrial mitigation strategies can partially mitigate air–ground interference, severe interference cannot be entirely eliminated. This paper presents an interference mitigation algorithm that is designed exclusively for downlink scenarios involving 5G-connected UAVs.

The objective of this research paper is to devise a new approach to power control that can mitigate interference in 5G networks, specifically addressing the challenges associated with the use of UAVs. To accomplish this, we put forward the utilisation of the DQL algorithm as a means of mitigating interference in power control.

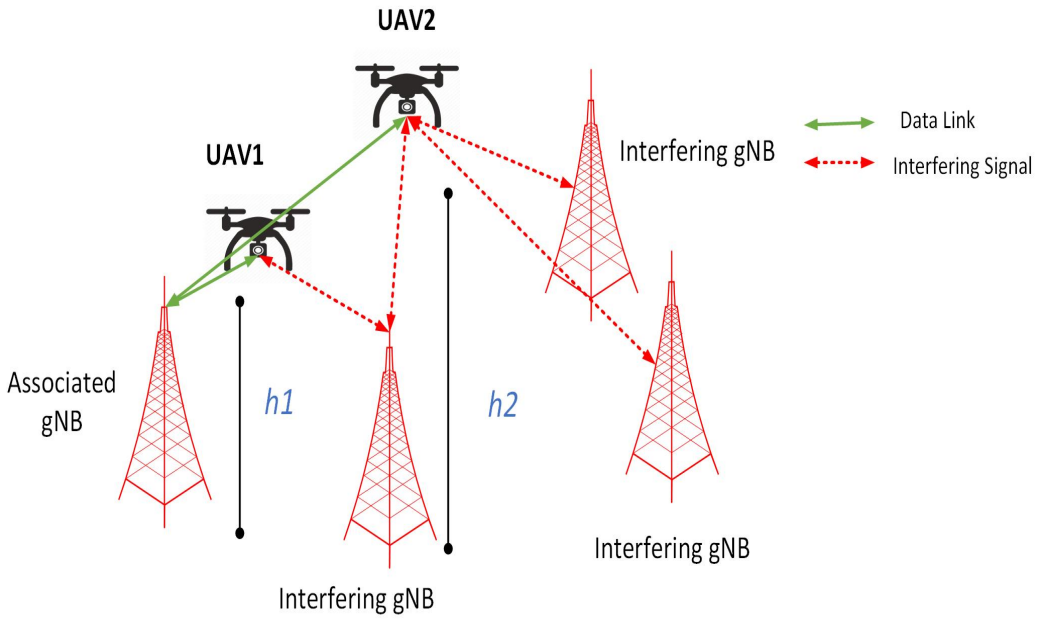

Figure 2. Interference in UAV networks.

4. 5G, UAV, and gNB Functionality

4.1. 5G Functionality

The utilization of 5G networks is regarded as advantageous for UAVs owing to their capacity to offer consistent connectivity while concurrently lowering cost and resource requisites. The 5G framework incorporates sophisticated radio access networks (RANs) and employs a plethora of technology enablers, such as beamforming, edge computing, network slicing, and others, to cater to an extensive array of wireless services. These advancements have a considerable influence on UAV communication, enabling greater flexibility and efficiency in operation.

The recently introduced 5G Network Core is outlined in the Third Generation Partnership Project (3GPP) standard [40] and is depicted in Figure 3, utilising a cloud SBA. This updated system encompasses all 5G-related functions and interactions, including session management, authentication, protection, and end-user traffic aggregation, which are briefly described below as defined in

1. Network Exposure Function (NEF): A network component known as the Network Exposure Function (NEF) serves as a bridge between 3GPP core network functionalities and external entities, including third parties and non-3GPP environments. NEF plays a vital role in ensuring the security of interactions when services or application functions (AFs) connect to 5G core nodes. It can be likened to a versatile intermediary, an Application Programming Interface (API) aggregation hub, or a translator that facilitates seamless integration into the 5G core network.
2. Network Function Repository Function (NRF): The 5G Network Function Repository Function (NRF) is a critical component of the 5G network architecture that acts as a central registry for all the network functions available in a 5G network. The NRF maintains a database of all the network functions that are available in a 5G network, including core network functions, service functions, and network slice functions. It also provides information about the location and availability of these functions, allowing other network functions to locate and access them when needed. Some of the key functions of the NRF include registering and storing information about all network functions available in the network. The NRF is an essential component of the 5G network, enabling the dynamic and flexible allocation of network resources to support a wide range of use cases and services. By providing a centralised repository for network functions, the NRF simplifies the process of locating and accessing

network resources, making it easier for network operators to deploy and manage their networks.

3. Network Slice Selection Function (NSSF): The 5G Network Slice Selection Function (NSSF) is a key component of the 5G network architecture that is responsible for selecting and assigning network slices to users based on their specific requirements and the availability of network resources. A network slice is a logical partition of the 5G network that is created to provide a specific set of services and capabilities tailored to the needs of a particular user or group of users. Network slices can be created for a wide range of use cases, including mobile broadband and mission-critical services. The NSSF plays a crucial role in enabling the dynamic and flexible allocation of network slices by performing user and device authentication and authorisation, network slice selection, and network slice management. Also, the NSSF allocates network resources to support the delivery of network slices, including RAN resources, core network resources, and transport network resources.
4. The Access and Mobility Management Function (AMF): It is a network function that is responsible for managing access and mobility for devices or users. It authenticates and authorises the device or user, manages access to the network, and ensures seamless mobility as the device or user moves between different network locations or coverage areas. The AMF also manages ongoing sessions for the device or user, monitoring QoS and ensuring that appropriate resources are allocated for the session. As a critical component of the 5G core network architecture, the AMF plays a crucial role in enabling seamless connectivity and mobility for 5G devices and users.
5. The Unified Data Management (UDM): It is a key component of the 5G network architecture that is responsible for managing user-related information, such as authentication and authorisation data, user preferences, and service subscriptions. The UDM is a centralised function that stores and manages all the user-related information that is needed by the 5G network, including information about the user's device, SIM card, and subscription. The UDM provides a unified view of user data to other network functions, allowing them to access the data they need to deliver services and manage network resources. The key functions of the UDM include authenticating users and authorising access to network resources based on their credentials and service subscriptions. The UDM provides policy control functions to manage network access, traffic routing, and other network resources based on user profiles and service requirements. The UDM manages subscriber data, including SIM card information, service subscriptions, and billing information. The UDM is a critical component of the 5G network architecture, providing a centralised repository for user-related information that enables other network functions to deliver customised and optimised services to users.
6. The Policy Control Function (PCF): The Policy Control Function (PCF) is a key component of the 5G network architecture that is responsible for managing policy decisions and enforcement in the network. The PCF acts as a centralised policy controller, defining and enforcing policies for network access, traffic routing, and other network resources. The PCF uses information from the UDM function to make policy decisions based on user profiles and service requirements. The PCF is an essential component of the 5G network architecture, providing a centralised function for policy control that enables operators to manage network resources and deliver customised services to users. By defining and enforcing policies for network access, traffic routing, and resource allocation, the PCF ensures that network resources are used efficiently and effectively, enabling operators to deliver high-quality services to their customers.

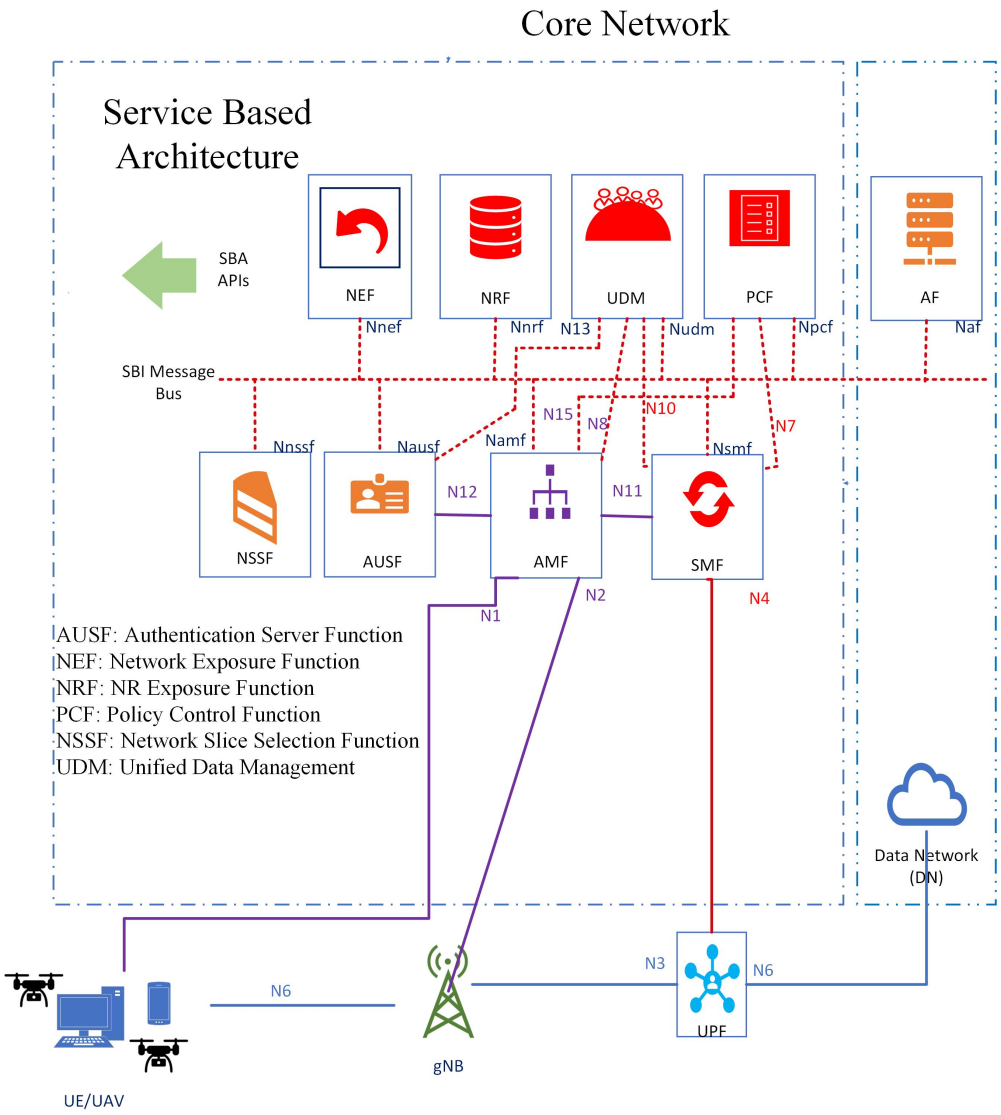

Figure 3. 5G-software-based architecture.

Thus, the SBA places a strong emphasis on the deployment of virtualised functions using the Multi-access Edge Computing (MEC) infrastructure as a fundamental architectural tenet. MEC is an essential element of 5G infrastructure, which seeks to enhance cloud computing by relocating applications from centralised data centres to the edge network in closer proximity to end-users. This strategy offers numerous benefits, including reduced latency, increased bandwidth, and real-time access to RAN data. The implementation of MEC is particularly important for the Command and Control (C2) link, which requires extremely low-latency communication. The proposed algorithm is implemented in a cloud-based architecture as part of the SBA infrastructure discussed above.

4.2. gNB Functionality

The primary component of the 5G-RAN architecture is the gNB, which stands for "Next Generation NodeB". The gNB provides the air interface for user devices to connect to the 5G network and manages the radio resources used for wireless communication. The flight controller unit reports SINR values, while the Global Positioning System (GPS) module reports coordinates. Ground controls primarily perform tasks such as controlling the UAV's attitude, displaying and manipulating payload data, planning missions, tracking the UAV's position, displaying routes on maps, positioning navigational targets, and maintaining communication links with other subsystems. According to the DQL algorithm proposed in

this work, the gNB receives coordinates and receives SINR values from the UAV to reduce interference. Figure 4 represents the system model used in this work, which shows the RAN network linked to the MEC architecture and core network.

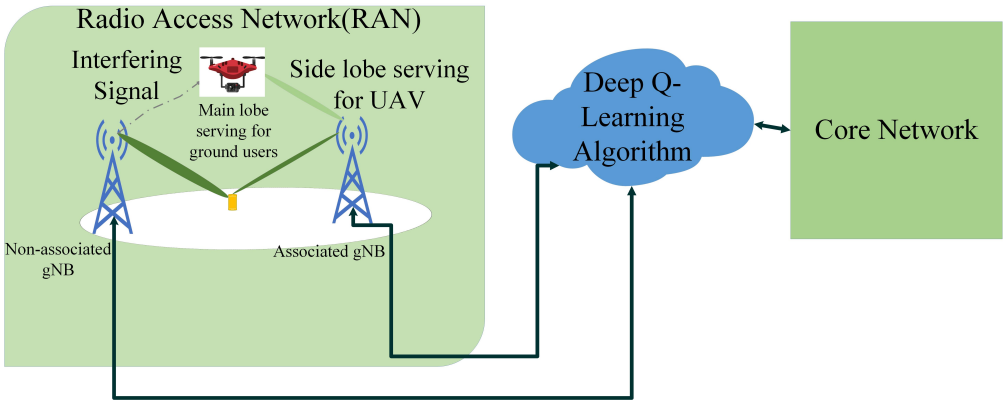

Figure 4. Interference mitigation system model.

MEC is a key technology in the 5G network architecture that enables the computation of traffic, processes, and services to be moved from centralised data centers to the edge of the network, closer to the end-users. By processing data and running applications closer to the RAN, MEC can significantly reduce latency, improve network performance, and enable new high-bandwidth, low-latency applications that were not possible with earlier mobile network technologies. Figure 5 depicts the MEC architecture as per the 3GPP standard [41]. It consists of a MEC orchestrator and MEC host. The MEC orchestrator interacts with the NEF, while the MEC host is deployed in the data network of the 5G system. The NEF can also be deployed at the edge to enable low-latency and high-throughput service access. The proposed DQL algorithm is implemented in the MEC host along with other applications on the application platform.

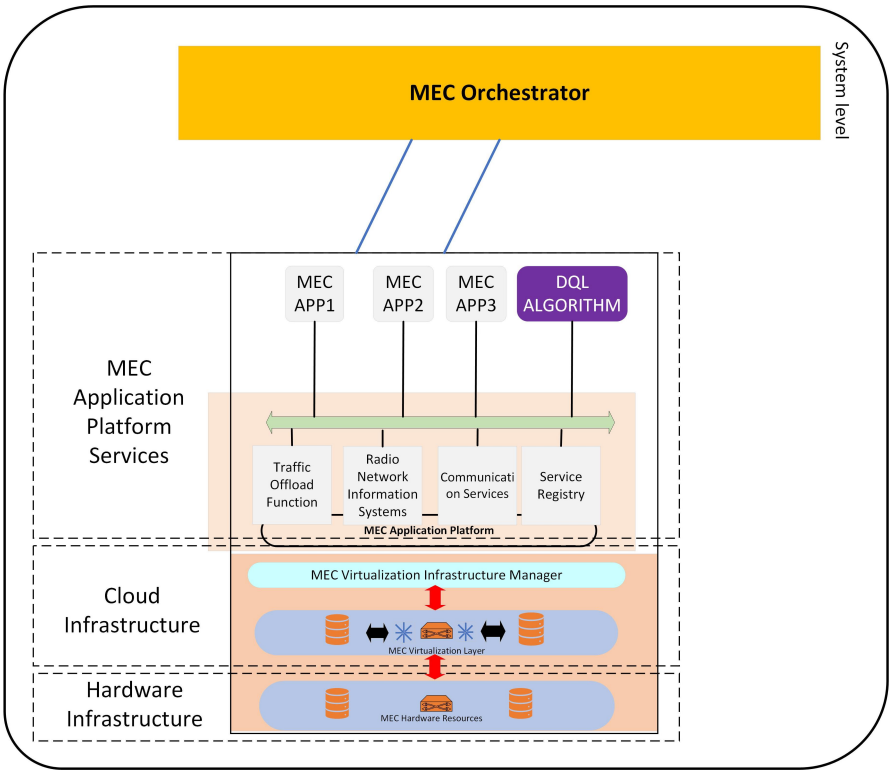

Figure 5. MEC cloud architecture.

Figure 6 shows the functional blocks of the system model from Figure 4. This is divided into three sections: (i) RAN: within the RAN, a UAV is being served by 1 gNB and receives an interfering signal from a non-associated gNB. The UAVs are primarily served by the side lobes of gNB antennas as the main lobes are down-tilted for optimal support to mobile devices. This also shows the structure of the gNB and its different components. (ii) Cloud location: the gNBs in RAN are linked to the cloud architecture, where the proposed DQL is deployed. The MEC architecture block was shown in Figure 5. It receives the SINR and location parameters from the gNB and computes the optimal power control commands and sends them back to the gNB. The gNB, in turn, sends these to the UAVs and thus interference is mitigated. Further, the cloud is linked to the core network. (iii) Core network: the core network is linked to the RAN via the cloud architecture and provides support through the virtualised functions as discussed in Section 4.1.

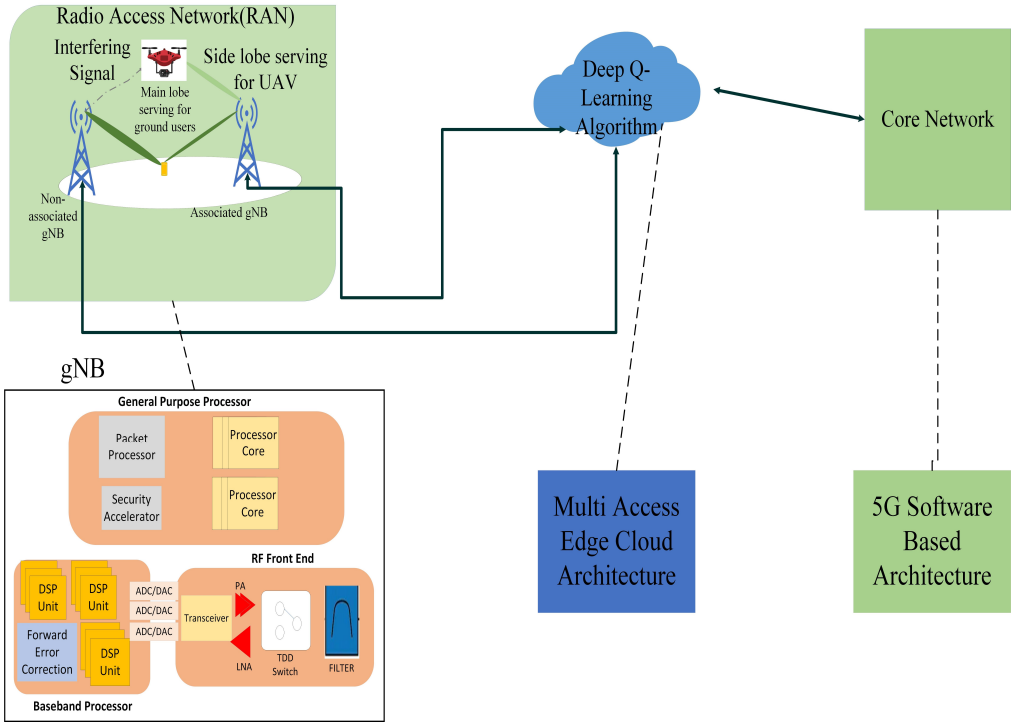

Figure 6. Interference mitigation system functional blocks.

This paper proposes a solution to the issue of interference in 5G networks caused by non-associated gNB for UAVs. The solution involves the implementation of a DQL algorithm within a cloud architecture. The algorithm is designed to enhance the SINR of UAVs and reduce interference by modifying the functions of the gNB. The decision making process takes place at a central location, where data from UAVs are transmitted over the backhaul to the central hub, which is then connected to the 5G core network.

5. Reinforcement Learning

Reinforcement learning (RL) is a specialised area within the realm of machine learning that concerns itself with determining appropriate actions to be taken by an intelligent agent in a given environment. The objective of this methodology is to maximise the cumulative reward obtained by the agent by selecting the most optimal action. It places emphasis on finding a balance between exploration and exploitation, thereby continually refining the agent's decision making capabilities. The agent must actively gather information about possible states, actions, transitions, and rewards. Unlike supervised learning, where evaluation is separate, in RL, the agent is evaluated while learning. The agent interacts with the environment through perception and action. The agent receives input in the form

of the current state of the environment and selects an action to alter the environment's state. The agent then receives a scalar reinforcement signal or reward based on the value of the state transition. The agent's goal is to maximise the long-run sum of the reinforcement signal's values by selecting actions that tend to increase the reward.

Table 2 provides definitions for various elements involved in the training phase that determine interaction between the agent and states and provide expected discounted reward value. Figure 7a illustrates inter-element interaction. The agent changes its state from s to $s^{'}$ by taking an action a in an environment and receives reward w for it.

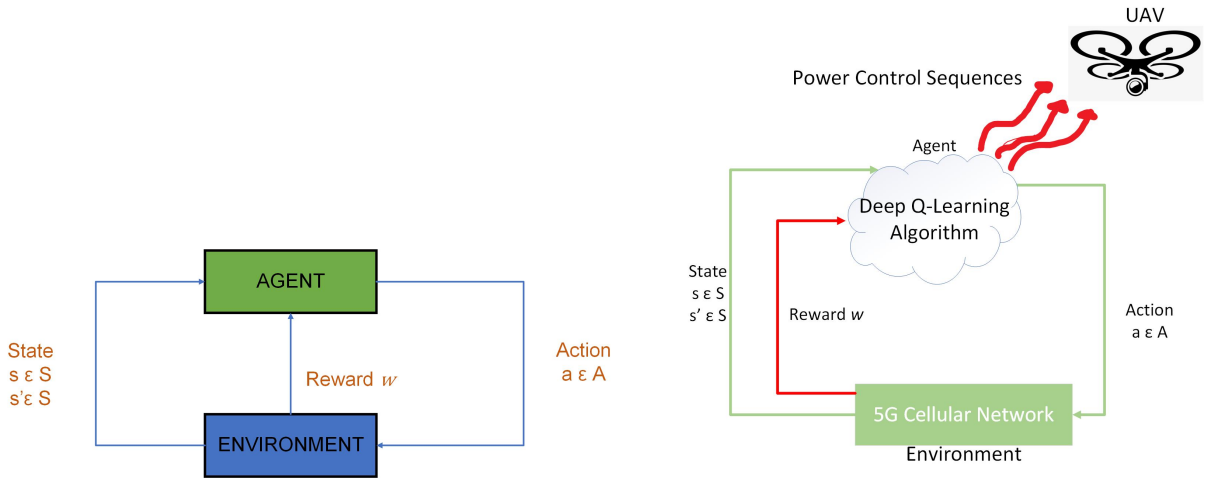

Figure 7. Agent–environment interaction. (**a**) RL elements; (**b**) DQL algorithm–agent–environment interaction.

In this work, the agent is the DQL algorithm, which exists in a given state s, i.e., a specific power value. It takes a suitable action, i.e., increase or decrease the power value, to optimise the SINR value. This interaction takes place in a 5G cellular environment. Finally, these commands are sent to the UAV. This is depicted in Figure 7b.

Table 2. Elements of reinforcement learning.

Elements	Definition
Observations	Observations are measurements of characteristics of the environment. They are expressed as $\mathcal{O}\epsilon\mathbb{R}^p$, where p is the no. of characteristics.
States	A state is defined as the discretised form of the observations at time step t, and is denoted as $s_t \epsilon$ S, where s_t represents the current state.
Actions	An action is a valid choice that can be made at a given time step t. It changes the state from s to the target state s'. It is denoted by $a_t \epsilon \mathcal{A}$.
Policy	The relationship between the state and the action is governed by a policy, denoted as $\pi(\cdot)$. The policy determines how the agent selects a new state by taking an action.
Rewards	After the agent takes 'a' when it is in 's' at t and moves to s', it receives a reward. The reward signal obtained is represented as $w_{s,s',a}[t;q]$.

5.1. Q-Learning

Q-learning is a commonly used reinforcement learning algorithm for making optimal decisions in Markov decision processes (MDPs). In Q-learning, the agent learns a Q-function that maps a state–action pair to a value representing the expected cumulative reward obtained by taking that action from that state and following an optimal policy thereafter. Thus, the Q-table is the mapping of state–action pairs with a Q-value relating to the reward. Table 3 illustrates a Q-table.

Table 3. Example of Q-table with state–action pairs mapped to Q-values.

State–Action	Q-Value
(S1, A0)	4
(S3, A4)	2
(S5, A1)	1

The Q-value is not calculated in a fixed manner; instead, it is implemented in the Q-table as an iterative approach. This is known as the training phase.

The Q-table is a fundamental component in the Q-learning algorithm, serving as a means of storing and updating information regarding the states, actions, and their associated expected rewards. It is implemented as a mapping between state–action pairs and their corresponding Q-values, which represent the estimated optimal future rewards of taking a particular action from a specific state.

The Q-learning algorithm works through the following steps:

Initialise Q-table: At the initiation of the Q-learning process, the Q-table is initialised to an array of zeros, indicating a lack of prior knowledge about the environment. As the agent interacts with the environment through trial and error, it updates its understanding of the state–action pairs and their corresponding Q-values, which it uses to optimise its future actions and maximise its expected reward.

Choose an action: To choose an action in Q-learning, the agent can use an exploration–exploitation strategy. During the exploration phase, the agent randomly chooses actions to gain information about the environment. During the exploitation phase, the agent chooses actions based on its current estimate of the optimal action-value function. One common exploration–exploitation strategy is the epsilon-greedy strategy. The agent chooses a random action with probability ϵ, and the optimal action (i.e., the action with the highest Q-value) with probability $1 - \epsilon$. As the agent gains more experience, it can gradually decrease the value of ϵ, which allows it to explore less and exploit more.

Update Q-table: The Bellman Equation is a mathematical expression that facilitates the determination of updates to the Q-table following each step taken by the agent. The equation effectively integrates the current perception of value with the anticipated optimal reward, which is premised on the selection of the optimal action as known at that moment. In a practical implementation, the agent assesses all feasible actions for a given state and chooses the state–action pair with the maximum Q-value. The Bellman Equation [31] is formulated as in (1).

$$Q_{\pi}^{\star}(s_t, a_t) := E_{s'}t\left[w_{s,s',a} + \gamma max_{a'} Q_{\pi}^{\star}(s', a') \middle| s_t, a_t\right] \tag{1}$$

Here, $\gamma : 0 < \gamma < 1$. s' is the next state. a' is the next action. These steps are repeated until the optimal solution is converged. This is also represented in Figure 8.

In this paper, Q-learning for interference mitigation is considered the baseline method against which the proposed algorithm is compared.

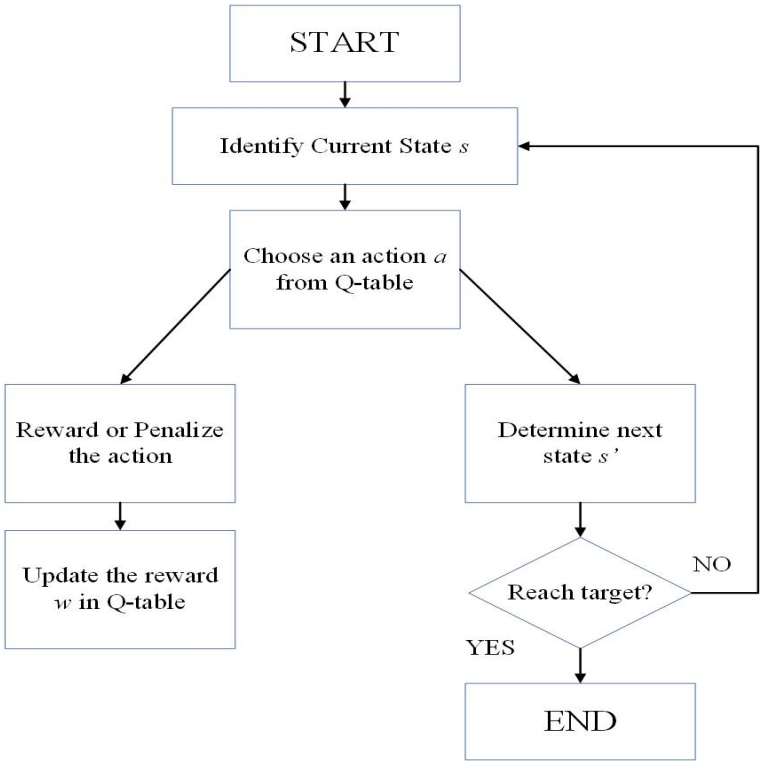

Figure 8. Representation of Q-table working.

5.2. Deep Learning

Deep learning algorithms use gradient descent optimisation and back-propagation to iteratively adjust the weights of the network and improve its predictions, with the goal of minimising the prediction error on the training data. The hierarchical representation learning in deep learning enables the model to learn increasingly complex features and representations, leading to its widespread use in numerous applications, including natural language processing, computer vision, and speech recognition. The depth and number of layers in deep learning models can vary, ranging from simple feed forward networks to complex recurrent neural networks and convolutional neural networks [42].

A neural network (NN) is an AI model composed of interconnected neurons or activations arranged in multiple layers. The NN transforms input data into output data based on a pre-established training dataset. This mapping is accomplished through the distribution of adjustable parameters known as weights across the various layers. The weights are refined through the use of the back-propagation algorithm, which minimises a loss function that gauges the deviation between the network's predictions and actual values. To reduce the loss function (or to optimise the back-propagation algorithm), gradient descent is used. In the mathematical theory of neural networks, universal approximation theory establishes the density of an algorithmically generated class of functions. Therefore, if (f(x)) is an arbitrarily complex function, then the neural network can approximately approach the solution of the function irrespective of its type. This implies that neural networks are capable of representing a wide variety of functions when given appropriate weights [43]. Activation functions help NNs differentiate between useful and irrelevant data points.

5.3. Neural-Network-Based Function Approximator

If the initial state–action value function $Q_\pi(s, a)$ is modified for every time instant 't', then it will converge to $Q^*_\pi(s, a)$ as $t \to \infty$. However, this is not easy to achieve. The primary cause for this is due to its implementation in a non-stationary environment. Real-world environments, such as the 5G-UAV integrated network considered in this paper, are often dynamic and may change over time. In such environments, the optimal policy itself may change, making it difficult for the algorithm to converge. Frequent updates of the state–

action values can exacerbate this issue as the learning process might not be able to keep up with the changes in the environment. To counteract this, we use a function approximator.

At a given time, a neural network with its weights is defined as $\varphi_t \in \mathbb{R}^{uv}$. Also, if $\varphi_t := vec(\varphi_t) \in \mathbb{R}^{uv}$ is defined, a function approximator $Q_\pi(s,a;\varphi_t) \approx Q_\pi{}^*(s,a)$ is constructed.

This neural-network-based function approximator forms the DQN. An important component of neural networks is activation functions. The sigmoid function, as shown in (2), is a popular choice [31] for the activation function. The sigmoid function is a mathematical function that maps any real-valued number to a value between 0 and 1. It is defined as

$$\sigma : x \hookrightarrow 1/(1+e^{-x}) \tag{2}$$

The deep network is trained by modifying φ for every t to reduce the mean-squared error loss represented by $K_t(\varphi_t)$:

$$\underset{\varphi_t}{\text{minimise}}\, K_t(\varphi_t) := E_{s,a}\left[(m_t - Q_\pi(s,a;\varphi_t))^2\right] \tag{3}$$

where

$m_t := E_{s'}\left[w_{s,s',a} + \delta max_{a'}\,(Q_\pi, a'; \varphi_{t-1})|s_t, a_t\right]$ is the approximate value of the function at t, when the current state is s and action a. Interacting with the environment in this manner and making a prediction, comparing it against the correct response, and suffering a loss $K_t(\cdot)$ is termed online training. The associated gNB relays the UAVs' feedback data to a central location for DQN training in online learning [31].

The dimension of the input layer is the no. of states $|\mathcal{S}|$, while the output layer is set to the no. of actions $|\mathcal{A}|$. A small depth is chosen for the hidden layer as it has a direct impact on the computational complexity. This is shown in Figure 9.

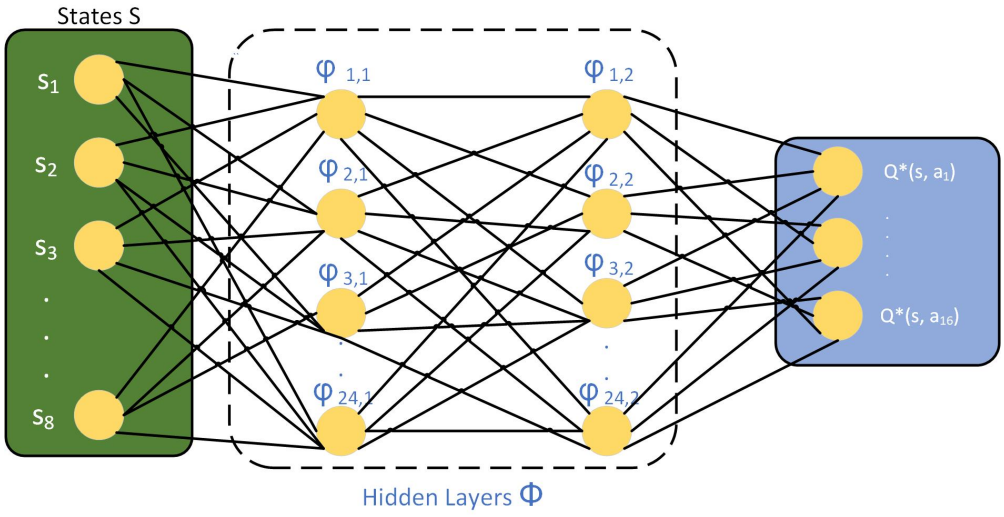

Figure 9. Structure of the DQN, with two hidden layers of dimension H.

5.4. Deep Reinforcement Training Phase

Stochastic Gradient Descent

During the training phase, the weights are subject to iterative modifications employing the Stochastic Gradient Descent (SGD) technique. SGD represents a variant of the gradient descent method, wherein a differentiable (or sub-differential) function is optimised through stochastic approximation. This is achieved by substituting the actual gradient, computed from the complete dataset, with an estimated gradient derived from a randomly selected minibatch of the dataset. By adopting this approach, particularly in optimization problems characterised by high dimensionality, the computational burden is mitigated, resulting in accelerated iterations.

During the training phase, the weights φ_t are subject to modification following each iteration of time, achieved through the utilisation of the SGD method on a selected mini-batch of data. It starts with an initial value of φ, which is chosen at random, and iterates and updates this value using an $\eta > 0$ step size as follows:

$$\varphi_{t+1} := \varphi_t - \eta \nabla L_t(\varphi_t) \tag{4}$$

The training is supported by replaying the experience from a replay buffer $\mathcal{D}$. $\mathcal{D}$ stores the different experiences i_t from different episodes at each time step. i_t is defined in (5)

$$i_t := (s_t, a_t, w_{s,s',a}[t;q], s'_t) \tag{5}$$

Subsequently, random samples were drawn from the dataset $\mathcal{D}$ and organised into minibatches. The training process is then conducted on these minibatches of data. This approach offers stability and mitigates the risk of convergence to local minima. As a result, the parameters used to generate the sample from $\mathcal{D}$ differ from the current parameters of the deep neural network.

The state–action value function $Q^*_\pi(s,a)$ estimated by the deep Q-network (DQN) is defined by Equation (1).

5.5. Policy Selection

Q-learning is a type of off-policy reinforcement learning algorithm, which allows for the creation of a nearly optimal policy even if actions are selected based on a random exploratory policy. As a result, a nearly greedy action selection policy is adopted, consisting of two modes.

- Exploration: In the beginning, when the agent is unaware of the most effective action, it tries different actions randomly to determine an effective action a_t
- Exploitation: Once the agent learns the various actions, it then chooses an action based on this knowledge to maximise the state–action value function $Q_\pi(s,a;\varphi_t)$.

According to this policy, where $\epsilon : 0 < \epsilon < 1$ is a hyperparameter, regarding the trade-off between exploration and exploitation:

(i) the agent carries out exploration with a probability ϵ.
(ii) $1-\epsilon$ probability is used for exploitation.

The trade-off results in this policy being referred to as ϵ-greedy action selection policy. This policy has a linear regret in t.

In the system model adopted in this work, it is assumed that UAVs move at speed v and the agent selects a_t from its s_t and receives $w_{s,s',a}[t;q]$ and finally moves to $s' := s_{t+1}$. An episode is the time period in which the agent and environment interact. If the target is achieved within the episode, then it is said to have converged.

In the proposed DQN implementation, the UAV coordinates are particularly kept track of. These coordinates are reported to the network and used for decision making to improve the network performance. Thus, UAV coordinates are part of the DRL framework.

5.6. Hyperparameter Tuning Using Random Search

Hyperparameter tuning is required for the proposed DQL algorithm to achieve better performance and stability during the training process. In this work, we use the random search technique to effectively tune the hyperparameters.

Random search is a simple and intuitive hyperparameter tuning technique widely used in machine learning and deep learning. Its primary goal is to find optimal hyperparameter configurations that maximise the performance of a model in a given task. Random search samples hyperparameter values randomly from predefined ranges. The process of random search involves defining a search space for each hyperparameter, typically specified by minimum and maximum values. During each iteration, the algorithm selects random values for each hyperparameter from their respective search spaces. These randomly

sampled hyperparameter configurations are then used to train the model. The random search method is implemented similar to [31] as the environments used in their work are similar to ours. Random search entails the following steps:

1. Define search space: This is a range of possible values to sample from for each hyperparameter, i.e., learning rate (α), discount factor (γ), and exploration rate (ϵ).
2. Initialise optimal hyperparameters: Before the random search begins, we initialise the hyperparameters to an initial value. In the proposed algorithm, a maximization problem is considered; therefore, the initial values are set to a very low value. After, this random search is performed.
3. Train the DQL agent with sampled hyperparameters: Using the sampled values obtained after the random search, the DQL agent is trained within the environment until convergence.
4. Evaluate the performance: Evaluate the agent's performance by running the agent in the environment for a fixed no. of episodes. Compute the average reward as given by expression in Table 2 as a performance metric.
5. Update the optimal hyperparameters: If the values of hyperparameters after random search are better than initial optimal values, then the optimal values are updated to current values.

The optimal hyperparameter values obtained from this method are provided in Table 4.

Table 4. RL parameters.

Parameter	Value
Discount Factor γ	0.995
Initial exploration rate ϵ	1.000
Number of States $\mathcal{S}$	8
Deep Q-Network width H	24
Exploration rate	0.995
Minimum exploration rate $\epsilon_{min}, \epsilon_{min}^{thresh}$	(0.15, 0.10)
No.of actions $\mathcal{A}$	16
DQN Depth	2

6. Existing Methods and Proposed Solution

6.1. Fixed Power Control (FPA)

Fixed power allocation (FPA) is a standard industry technique for interference mitigation and is, therefore, used as a baseline algorithm. It assigns a specific value for the transmit signal power. FPA does not use interference coordination. The total transmission power is uniformly allocated across all physical resource blocks (PRBs), resulting in a constant power level. The gNB retains its power level and adjusts the code schemes and modulation techniques. This is termed link adaptation. The link is adapted based on the measurements reported by the UAV back to gNB. The link is adapted either based on periodic or aperiodic measurements because the transmit power is fixed, resulting in a higher effective SINR. There are no measurement reports sent to the interfering gNB [44]. The total power is simply divided among the PRBs:

$$P_{\mathrm{TX},b}[t] := P_{\mathrm{gNB}}^{\max} - 10\log N_{\mathrm{PRB}} + 10\log N_{\mathrm{PRB},b}[t] \quad \text{(dBm)} \tag{6}$$

6.2. Tabular RL

Tabular Q-learning is a solution used for interference mitigation. In this method, the state–action value function is denoted by $Q_\pi(s_t, a_t)$. The Q-learning update process is defined by the learning rate. This is due to the fact that, in the tabular setting, the Q-table must store the Q-values for every possible state–action pair. In practical applications, the state and action spaces can be very large, resulting in a large and possibly infinite

size of the Q-table. To ensure convergence, the Q-table must be updated continuously with new experiences, and all state–action pairs must be visited and updated an infinite number of times. However, in practice, this is not feasible, and convergence may be slow or never reached. This is why the tabular setting of Q-learning is more suited to problems with smaller state spaces. Additionally, it also requires a non-trivial initialisation of the $Q \epsilon \mathbb{R}^{|S| \times |A|}$ table to avoid longer convergence times. Therefore, computationally, it suits problems with smaller state spaces. Q-learning has no neural network involvement. The Q-learning update analog of (1) is defined as

$$Q_\pi(s_t, a_t) := (1 - \alpha) Q_\pi(s_t, a_t) + \alpha \left(w_{s,s',a} + \delta \max_{a'} Q_\pi(s', a') \right) \tag{7}$$

In this context, $Q_\pi(s_t, a_t) := [Q]_{s_t, a_t}$. The learning rate, $\alpha > 0$ determines the level of aggressiveness of the Q-learning update and sets how much weight is given to prior experience. The table-based approach is suitable for problems with small state spaces, allowing for the storage of the table Q.

6.3. Proposed Solution

The primary objective of the optimization problem is to maximise the sum of $\delta^i[t]$ for all i in the set $\{1, 2, \ldots, L\}$, which represents the gNBs. In mathematical terms, the objective function is

$$\begin{aligned} &\underset{P_{\mathrm{TX},i}[t],\ \forall i}{\text{maximise}} \sum_{i \in \{1,2,\ldots,L\}} \delta^i[t] \\ &\text{if } P_{\mathrm{TX},i}[t] \in \mathcal{P}, \quad \forall i, \\ &\qquad \delta^i[t] \geq \delta_{\text{target}}. \end{aligned} \tag{8}$$

Here, $P_{\mathrm{TX},i}[t]$ represents the transmission power for the ith gNB at time t. The objective is to find the values of$P_{\mathrm{TX},i}[t]$ that maximise the sum of $\delta^i[t]$.

The optimization variables are $P_{\mathrm{TX},i}[t]$ for all i gNBs. These variables represent the transmission power levels for different gNBs at a given time t. The goal is to find the optimal values for these transmission power levels. To ensure that the optimization remains feasible and practical, there is a constraint on the transmission power levels. The transmission power $P_{\mathrm{TX},i}[t]$ for each gNB i must be chosen from a predefined set $\mathcal{P}$. The second constraint is imposed on the values $\delta^i[t]$. Each $\delta^i[t]$ must be greater than or equal to a predefined minimum threshold value δ_{target}. The δ values at each gNB ß come from the UAV measurement reports.

The i-th gNB tries to solve this problem and, as a result, finds the optimal value for $P_{\mathrm{TX},l}$. The proposed DQL algorithm solves (8) at a central cloud location. Algorithm 1 is based on the DQL approach. It enables power control without the need for the UAV to transmit explicit power control commands, leading to lower computational overhead compared to tabular Q-learning for a given number of states and network depth of the DQN network [31].

The flow diagram for the proposed Algorithm 1 is shown in Figure 10. The main steps can be summarised as follows:

1. Choose an optimising action.
2. Choose a power control action.
3. Evaluate the impact on effective SINR $\delta^l_{eff}[t]$.
4. The action is rewarded based on the distance from δ_{target} or δ_{min}; for instance, a higher reward could be given to an agent that is in closer proximity to the δ_{target}, while a lower reward would be given to an agent that is closer to the δ_{min}.
5. The DQN is trained by utilising the produced outcomes.

The reported SINR from the UAVs is fed as input to the DQL algorithm. Firstly, the current states are observed. The algorithm starts with a higher value of ϵ, where the system selects random actions and learns about the best rewards. As it progresses, the ϵ value decays and the system keeps choosing the action with the highest reward. After a reward is received, the effective SINR δ_{eff} is calculated and checked against the minimum SINR δ_{min}. If $\delta_{\text{eff}} < \delta_{\text{min}}$, then the episode is aborted; otherwise, the next state is observed and stored in the experience buffer $\mathcal{D}$. From $\mathcal{D}$, the data are minibatched and SGD is performed on it. The results are used to update the DQN and the states are updated to s'. The minibatching and the consequent updating of states via back-propagation occur within the neural network as shown earlier in Figure 9. Finally, it is checked if $\delta_{\text{eff}} \geq \delta_{\text{target}}$, where δ_{target} is the target SINR. If it is, then the power control sequence commands are generated, or else the algorithm repeats until the target SINR is achieved. The pseudocode is shown in Algorithm 1 below:

Algorithm 1 Deep Q-Learning Algorithm for UAV interference mitigation

I/P: Downlink δ_{eff} reported by the UAVs
O/P:Power Control Commands to optimise received δ

Initialise states, time, replay buffer $\mathcal{D}$ and actions until convergence or termination
until $t \geq T$
Increment t
Current state observed is s_t
$\epsilon := \max(\epsilon \cdot d, \epsilon_{min})$
Sample r $r \leq \epsilon$
a_t ϵ $\mathcal{A}$ is randomly selected
$a_t = \arg\max_{a'} Q_\pi(s_t, a^{'}; \varphi_t)$
Calculate $\delta^l_{eff}[t]$ and $w_{s,s',a}[t;q]$ $\delta^l_{eff}[t] < \delta_{min}$
$w_{s,s',a}[t;q] := w_{min}$
Terminate episode
Next state s′ is observed
Save $i[t] \triangleq (s_t, a_t, w_{s,s',a'}, s^{'})$ in $\mathcal{D}$
Minibatch experience sample from $\mathcal{D}$: $i_j \triangleq (s_t, a_t, w_j, s_{j+1})$
$\gamma_i := w_j + y_{max_{a'}} Q_\pi(s_{j+1}, a^{'}; \varphi_t)$ is set
Stochastic Gradient Descent is performed on $(y_i - Q_\pi(s_j, a_j; \varphi_t)^2)$, and determine φ^*
Update $\varphi_t := \varphi^*$ in the DQN and calculate loss L_t
$s_t := s^{'}$ $\delta^l_{eff}[t] \geq \delta_{target}$
$w_{s,s',a}[t;q] := w_{s,s',a}[t;q] + w_{max}$
end

DQL is an efficient solution to the problem of interference as it does not require channel CSI (unlike traditional link adaptation techniques) to calculate SINR. Additionally, it also reduces the requirement for UAVs to provide feedback to the gNB. Moreover, DQL also decreases the need for UAVs to send feedback to the gNB. Currently, as per the methods documented in existing works, UAVs need to report their CSI, leading to an increase in overhead and complexity due to the large amount of information (in the form of CSI) being transmitted in systems. The proposed DQL algorithm offers a more streamlined communication mechanism between UAVs and gNBs. The UAVs are only required to transmit their SINR and coordinates, leaving the agent in charge of power control. This stands in contrast to conventional practices in the industry, which usually necessitate the transmission of specific commands for power control to both the associated and interfering gNBs. The associated gNB need only transmit power control commands to the UAV, thus reducing the amount of transmitted information and simplifying the overall communication process. Table 5 summarises the symbols used.

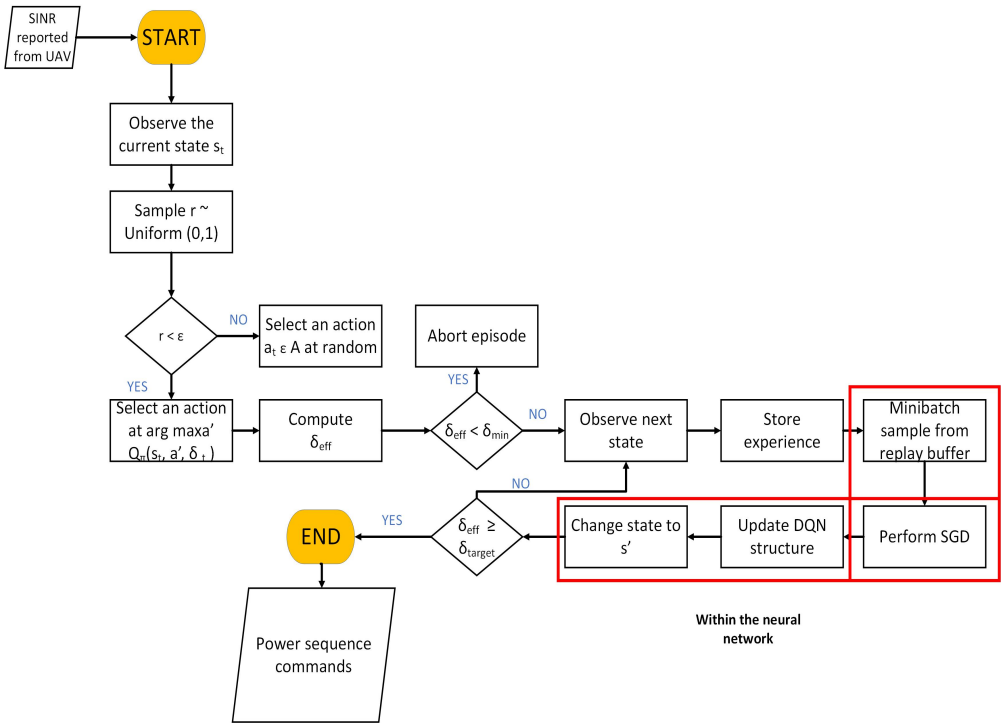

Figure 10. Deep Q-learning for interference mitigation flow diagram.

Table 5. List of notations.

Notation	Description
t	Time
s_t	States
ϵ	Exploration rate
d	Exploration rate decay
a_t	Actions
r	Data Sample
L	No. of associated gNBs
B	No. of interfering gNBs
i	Total no. of gNBs
h	No. of UAVs
Q_π	State–Action Value Function
φ_t	Neural network weights
$i[t]$	experience
δ	SINR
δ_{eff}	Effective SINR
δ_{target}	Target SINR
δ_{min}	Minimum SINR
δ_{thresh}	Threshold SINR
SGD	Stochastic Gradient Descent
P_{TX}	Transmit Power
$\mathcal{P}$	Set of transmit powers
$\mathcal{D}$	Experience Replay buffer
y_l	Received signal
$\mathbb{E}$	Estimated function
H	Neural Network Width

7. Network, System, and Channel Model

7.1. Network Model

In this scenario, a cellular network with $\mathcal{L}$ gNBs is analysed. Each network consists of one associated gNB and at least one interfering gNB, and the UAVs' association with their

associated gNB is based on their distance. A UAV can only be served by one gNB at a time, so the cell radius can be expressed as $r > R/2$.

7.2. System Model

Under the given network model and considering a multi-antenna setup, with each gNB equipped with M antennas and each UAV having a single antenna, the signal received at the UAV from the $\mathcal{L}$-th gNB can be expressed as

$$y_l = c_{l,l}^* h_l x_l + \sum_{b \neq l} c_{l,b}^* h_b x_b + n_l \tag{9}$$

Here, x_l, $x_b \, \epsilon \, \mathbb{X}$ are transmitted signals from the $\mathcal{L}$th and $\mathcal{B}$th gNBs, and it meets the power constraint

$$\mathbb{E}[|x_l|^2] = P_{\mathrm{TX,l}} \tag{10}$$

The vectors $c_{l,l}$, $c_{l,b}$, and $M \times 1$ are channel vectors representing the connection between the UAV and the $\mathcal{L}$-th gNB, and the connection between the UAV and the $\mathcal{B}$-th gNB, respectively. Finally, n_l is drawn from a complex normal distribution with zero mean and variance of σ_n^2, representing the noise received by the UAV.

The first term in Equation (9) represents the desired signal received by the UAV, while the second term represents the interference received from non-associated gNBs.

Each gNB $\mathcal{L}$ transmits power $P_{\mathrm{TX,l}} \, \epsilon \, \mathcal{P}$ from the set of all possible power values, denoted by $\mathcal{P}$. This set is defined such that the possible transmit power is a power offset.

7.3. Channel Model

In order to delineate the conditions of signal propagation, the concept of LoS probability is introduced to quantify the likelihood of the LoS component's presence. The 3GPP channel model [45] provides specific LOS probability models, with antenna heights of 3 m for indoor scenarios, 10 m for Urban Micro (UMi) environments, and 25 m for Urban Macro (UMa) settings. With respect to this work, the UMa scenario is chosen. This can be written as

$$P_{rLOS} = \begin{cases} 1 & , d_{3D-\mathrm{out}} \leq 18\mathrm{m}, \\ \left[\frac{18}{d_{3D-\mathrm{out}}} + \exp\left(\frac{-d_{3D-\mathrm{out}}}{63}\right)\left(1 - \frac{18}{d_{3D-\mathrm{out}}}\right)\right], & \\ & , 18\mathrm{m} < d_{3D-\mathrm{out}} \\ \left[1 + C^{'}(h_{UT})\frac{5}{4}\left(\frac{d_{3D-\mathrm{out}}}{100}\right)^3 \exp\left(\frac{d_{3D-\mathrm{out}}}{150}\right)\right] & \end{cases} \tag{11}$$

where

$$C^{'}(h_{UT}) = \begin{cases} 0, & h_{UT} \leq 13\mathrm{m} \\ \left(\frac{h_{UT}-13}{10}\right)^{1.5}, & 13\mathrm{m} < h_{UT} \leq 23\mathrm{m} \end{cases} \tag{12}$$

Pathloss presents a strong correlation with the channel modelling and is influenced by distance. Since UAVs operate in a three-dimensional environment, the pathloss is given by

$$PL_1 = 28 + 22\, log_{10}(d_{3D}) + 20\, log_{10}(f_c) \tag{13}$$

where d_{3D} is the location of the UAVs 3D distances and f_c is the operating frequency.

This algorithm adopts a narrow-band geometric channel model, as described in [12]. With this model, the downlink channel from gNB $\mathcal{B}$ to the UAV in gNB $\mathcal{L}$ can be expressed as

$$c_{l,b} = \frac{\sqrt{M}}{\rho_{l,b}} \sum_{p=1}^{N_{l,b}^{p}} \alpha_{l,b}^{p} a^{*}(\varphi_{l,b}^{p}) \tag{14}$$

where
$\alpha_{l,b}^{p}$: path gain of the p-th path;
$\varphi_{l,b}^{p}$: angle of departure(AoD);
$a(\varphi_{l,b}^{p})$: array response vector associated with AoD;
$N_{l,b}^{p}$: no. of channel paths.

This model accounts for both LOS and NLOS scenarios. For the LOS case, $N_{l,b}^{p} = 1$ is assumed.

The received power at the UAV is measured over a set of PRBs at a given time t and is given by

$$P_{\mathrm{UAV}}^{l,b}[t] = P_{\mathrm{TX},b}[t] \left| c_{\mathrm{l,b}}^{*}[t] h_b[t] \right|^2 \tag{15}$$

where $P_{\mathrm{UAV}}^{l,b}[t]$ is received downlink power, $P_{\mathrm{TX},b}[t]$ is power from gNB $\mathcal{B}$.

Some additional losses are also suffered due to penetration and oxygen absorption.

8. Simulation Set-Up and Results

In the system model considered in this paper, a UAV is operating within the range of 2 gNBs in a 5G-RAN network. One gNB is associated with the UAV and the other one is the interfering gNB. The UAV sends its measured SINR to the gNB, which is then served as the input to the DQL algorithm. The SINR received by the UAV served in gNB $\mathcal{L}$ at time t is computed as follows

$$\delta^{\ell}[t] = \frac{P_{\mathrm{TX},\ell}[t] |c_{\ell,\ell}^{*}[t] h_{\ell}[t]|^2}{\sigma_n^2 + \sum_{b \neq \ell} P_{\mathrm{TX},b}[t] |c_{\ell,b}^{*}[t] h_b[t]|^2} \tag{16}$$

This is the received SINR that will be optimised by the proposed DQL, which is located in the cloud, linked to the RAN. The output of the DQL is then fed back to the gNB. The gNB then sends these commands back to the UAV, which optimises the SINR and mitigates interference.

8.1. Simulation Setup

The network, system, and channel models have been detailed in prior descriptions. The UAVs are in motion at a velocity of 'v' and are subject to the impact of log-normal shadow fading and small-scale fading on their movement. The radius of the cell is denoted as 'r', and the inter-site distance is established as R = 1.5r. The channel conditions of the UAVs are influenced by a probability of LoS denoted as p_{LOS}. The remaining parameters are presented in Table 4. The target effective SINRs for the objective are specified as

$$\begin{aligned} \delta_{\mathrm{target}} &:= 3\,\mathrm{dB}, \\ \delta_{\mathrm{target}}^{\mathrm{thresh}} &:= \delta_0^{\mathrm{thresh}} + 10 \log M\,\mathrm{dB} \end{aligned} \tag{17}$$

where δ_0^{thresh} is the established SINR. If the δ_0^{thresh} drops below a minimum of -3 dB, the episode is considered to have concluded and the session cannot proceed any further.

The hyperparameters used to tune the RL-based model are shown in Table 4. Table 6 summarises the radio parameters.

The simulated states $\mathcal{S}$ are set up as

$$\begin{aligned}(s_t^0, s_t^1) &:= \text{UAV}_\ell(x[t], y[t]), (s_t^2, s_t^3) := \text{UAV}_b(x[t], y[t]),\\ s_t^4 &:= P_{\text{TX},\ell}[t], \quad s_t^5 := P_{\text{TX},b}[t],\\ s_t^6 &:= \text{h}_n^\ell[t], \quad s_t^7 := \text{h}_n^b[t],\end{aligned} \tag{18}$$

where (x,y) are the Cartesian coordinates (i.e., longitude and latitude) of the given UAV.

Table 6. Radio environment parameters.

Parameter	Value
gNB power $P_{gNB}^{\max}$	46 dBm
Cellular Geometry	Hexagonal
Antenna Gain (TX, thresh)	(11, 3) dBi
Probability of LOS (P_{LOS}, $P_{\text{LOS}}^{\text{thresh}}$)	(0.9, 0.8) [45]
Downlink Frequency	4.7 GHz
Cell Radius r	150 m
UAV Antenna Gain	0 dBi
Inter-site distance R	225 m
Number of Multipaths N_p	15
Average UAV Speed v	(10, 20, 20) m/s
Frame Duration	20 ms

By leveraging the cardinality of $\mathcal{P}$, which scales in the order of 2, the set of actions $\mathcal{A}$ can be derived. Specifically, a register a is utilised to facilitate a binary encoding of the available actions. With the aid of masks, bitwise-AND operations, and shifting, the power control commands can be obtained. The code segment as shown in Table 7 is adopted:

When $q = 0$:

Table 7. Code for power control.

Code	Description
a[0, 1] = 00	power is reduced by 3 dB for gNB b
a[0, 1] = 01	power is reduced by 1 dB for gNB b
a[0, 1] = 10	power is increased by 1 dB for gNB b
a[0, 1] = 11	power is increased by 3 dB for gNB b
a[2, 3] = 00	power is reduced by 3 dB for gNB l
a[2, 3] = 01	power is reduced by 1 dB for gNB l
a[2, 3] = 10	power is increased by 1 dB for gNB l

It can be inferred from above that $\mathcal{P}$ = (±1, ±3) dB offset from the transmitter power. The following factors affected the choice of these offset values:

1. It conforms to the standards [46] that select integers for power value increments.
2. It maintains the non-convexity in (8) because it keeps the constraints discrete. Actions that increase or reduce gNB transmit powers are implemented as in [30].

A function $p(\cdot)$ is defined that returns an element from $\mathcal{P}$ depending on the selected code. In this work, it is considered that p(00) = −3, p(01) = −1, p(10) = 1, p(11) = 3. The received SINR resulting from the above actions can, therefore, be written as

$$\delta^{\text{b}}_{\text{a}_{[0]},\text{a}_{[3]}} \text{ for gNB b} \tag{19}$$

$$\delta^{\text{l}}_{\text{a}_{[1]},\text{a}_{[2]}} \text{ for gNB l} \tag{20}$$

We thus write the reward as

$$w_{s,s',a}[t;q] := \Big(\mathrm{p}(a_{[0,1]}[t]) - \mathrm{p}(a_{[2,3]}[t])\Big)$$
$$(1-q) + \left(\delta^{b}_{a_{[0]}[t],a_{[3]}[t]} + \delta^{\ell}_{a_{[1]}[t],a_{[2]}[t]}\right) q \tag{21}$$

where $q = 0$.

The agent is rewarded the most when it triggers a power control command. The episode is aborted when the constraint imposed on (8) becomes inactive. In this scenario, the agent receives a reward $w_{s,s',a}[t;q] := w_{\min}$ along with one of the following:

- A penalty is imposed by awarding the agent with the minimum possible reward $w_{\min}$. This occurs when the SINR falls below $\delta_{\min}$.
- A maximum reward w_{max} is imposed when δ_{target} is achieved.

In the simulations, the following steps are taken:

- Minibatching is performed with a sample size of $N_{\text{mb}} = 32$ training samples.
- The width of the DQN can be found as per [47]. Thus, (22) is written as

$$H = \sqrt{|A|+2)N_{\text{mb}}} \tag{22}$$

Substituting for $|A|$=16 and N_{mb}, we obtain H=24.

The following performance measures are used to benchmark our algorithms:

1. *Convergence (ζ):* Defined as the episode at which the δ_{target} is satisfied over the duration T for all UAVs. It is expected that, as the no. of UAVs h increases, ζ will increase. For several random seeds, an aggregated percentile convergence episode is taken.
2. *Runtime*: Determining the runtime complexity of the proposed DQL algorithm is difficult due to the absence of guarantees for convergence and stability. Therefore, runtimes from simulation per UAV are considered.
3. *Coverage:* A complement cumulative distribution function (CCDF) of δ^{ℓ}_{eff} is built by changing the random seed and performing the simulation in iterations, effectively modifying the way the UAVs drop from the network.

The proposed DQL overcomes the shortcomings of FPA by implementing an adaptive power control scheme that can accommodate the changing interfering conditions. Tabular Q-learning relies on the initialised state–action value function, which is fixed. This results in the size of the Q-table becoming quite large and therefore taking longer times to find the optimal solutions, thus resulting in longer convergence time. However, in the proposed DQL, the neural network implementation eliminates the need for a Q-table, instead updating the optimal actions in real time.

8.2. Results

Considering the episode with the highest possible reward, Figure 11 shows the plot of the Cumulative Complement Distribution Function (CCDF) of δ_{eff}, shown for the three algorithms FPA, tabular Q-learning, and the proposed DQL.

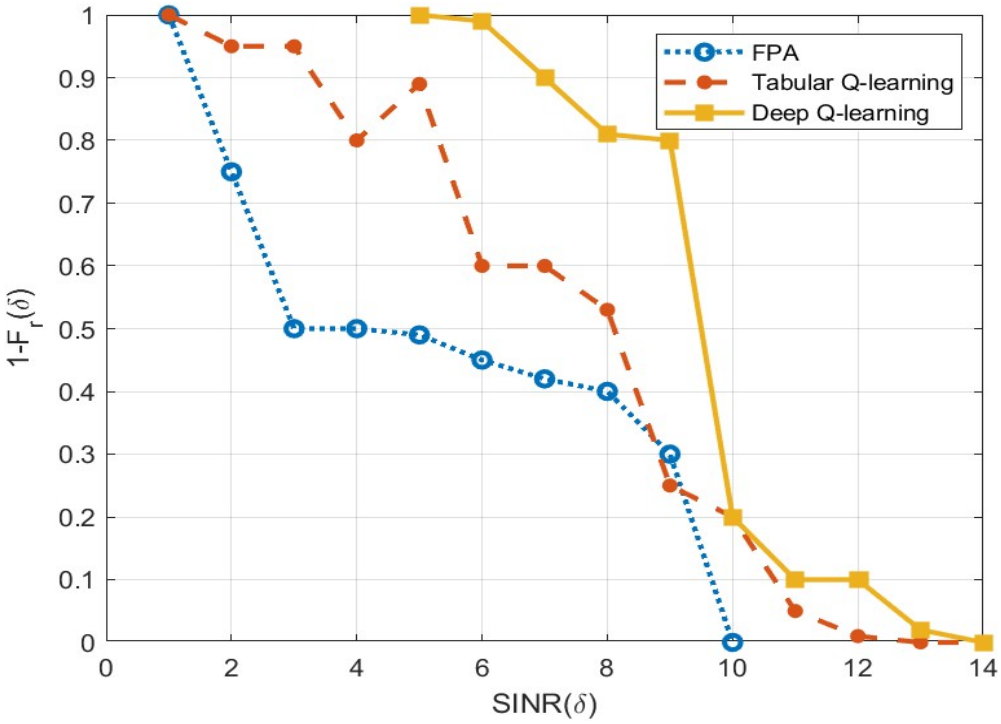

Figure 11. CCDF vs. δ_{eff} for FPA, tabular Q-learning, and DQL.

As anticipated, the FPA strategy produces the least optimal performance due to its lack of power control or interference coordination. Tabular Q-learning exhibits slightly improved results over FPA, yet it still falls short when compared to the addition of power control mechanisms to the gNBs, under-performing FPA at approximately $\delta_{eff} = 9$ dB. On the other hand, the proposed DQL algorithm outperforms the other two algorithms, achieving better convergence to a solution. Unlike the tabular implementation, the convergence of DQL is not affected by the initial state–action value function. As δ_{eff} approaches 13 dB, UAVs are situated in closer proximity to the gNB, resulting in comparable performance among all power control algorithms.

We also analyse the link performance for each of the algorithms at different heights of the UAV. As explained in the previous sections, when the altitude increases, the interference also increases.

The simulation is run for a duration of 120 s, with the UAVs flying at three different altitudes: 50 m, 100 m, and 120 m. To analyse the link performance, SINR and latency are compared for the different heights and the three different interference mitigation schemes.

8.2.1. UAV @ 50 m

In this section, we discuss the results when the UAVs are flying at a height of 50 m. This is a relatively lower altitude; therefore, the number of interfering elements is relatively lower.

Figure 12 shows SINR values for UAV @ 50 m. When no link adaptation is employed, i.e., when the FPA method is used, the values vary from 6.52 dB to 7.15 dB. This is a relatively steady performance; however, the SINR values are the lowest compared to Q-learning and DQL. This is expected as it is a lower altitude and the interference conditions do not change drastically, except for a brief 27 s peak between the 26 s and 53 s marks. Q-learning starts with a higher value of 9.7 dB and even goes up to 11.6 dB with an average value of 10.45 dB. This is indicative of good performance for the link; however, after about 95 s, it suffers a drastic dip in performance in addition to presenting erratic patterns till the end of the simulation. This can be seen in Figure 12 as the annotated text in the red circle. When the simulation progresses, the Q-table becomes large and achieving the optimal solution becomes difficult. This is also reflected in the link latency as shown in Figure 13.

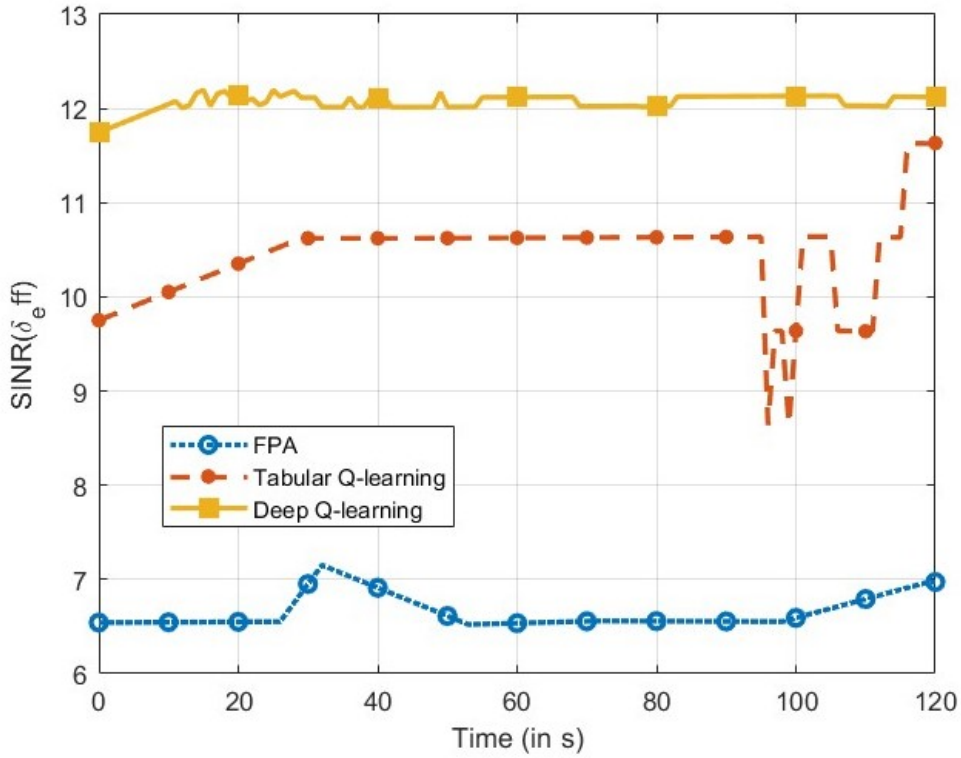

Figure 12. SINR vs. time @ 50 m.

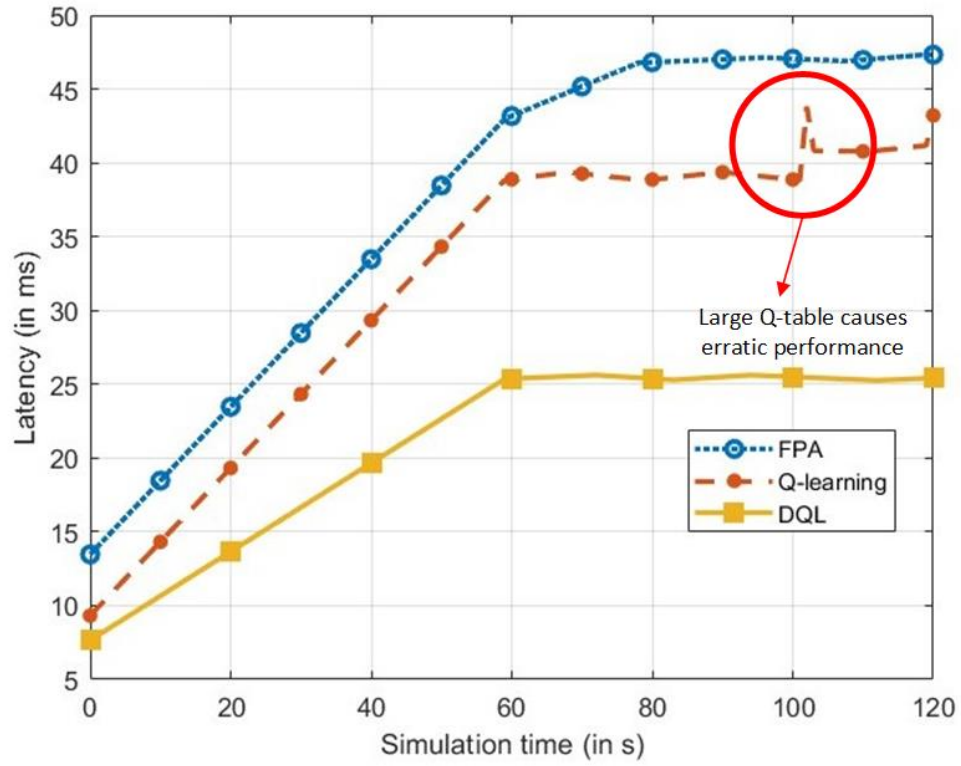

Figure 13. Latency at 50 m.

Finally, DQL performs the best with average, minimum, and maximum values of 12.06 dB, 11.74, and 12.19 db, respectively, which is indicative of steady performance with high SINR values. It also shows a really low latency value of 25.61 ms. Consequently, interference is best mitigated by DQL amongst the three methods.

8.2.2. UAV @ 100 m

In this section, we discuss the results when the UAV is flying at a height of 100 m. This is a higher altitude; therefore, the number of interfering elements is relatively higher.

Figure 14 shows SINR values for UAV @ 100 m. When no link adaptation is employed, i.e., when the FPA method is used, the values vary from 3.45 dB to 7.16 dB. This represents an unstable link as the SINR values vary drastically. For example, at about the 32 s mark, there is an increase to 6.95 dB from 6.5 dB, but, from thereon, it drops sharply to reach its minimum possible value. It increases again and does not stay constant. As mentioned above, this is primarily because the FPA algorithm cannot take into account the changing interference conditions, and, at a higher altitude of 100 m, the increase in interfering elements leads to a change in the interference conditions. This also coincides with the latency values, which see delays of up to 55 ms, as shown in Figure 15.

Q-learning starts with a lower value of 7.74 dB as compared to its 50 m performance. It maintains a steady performance value of close to 8.8 dB till about 88 s. This is indicative of good performance for the link; however, after about 89 s, it suffers a drastic dip in performance and undergoes a series of peaks and troughs till the end of simulation. At one point, specifically at the 104 s mark, it falls even below the FPA values. This can be seen in Figure 14 as the annotated text in the red circle. When the simulation progresses, the Q-table becomes large and achieving the optimal solution becomes difficult. This is also reflected in the link latency as shown in Figure 15.

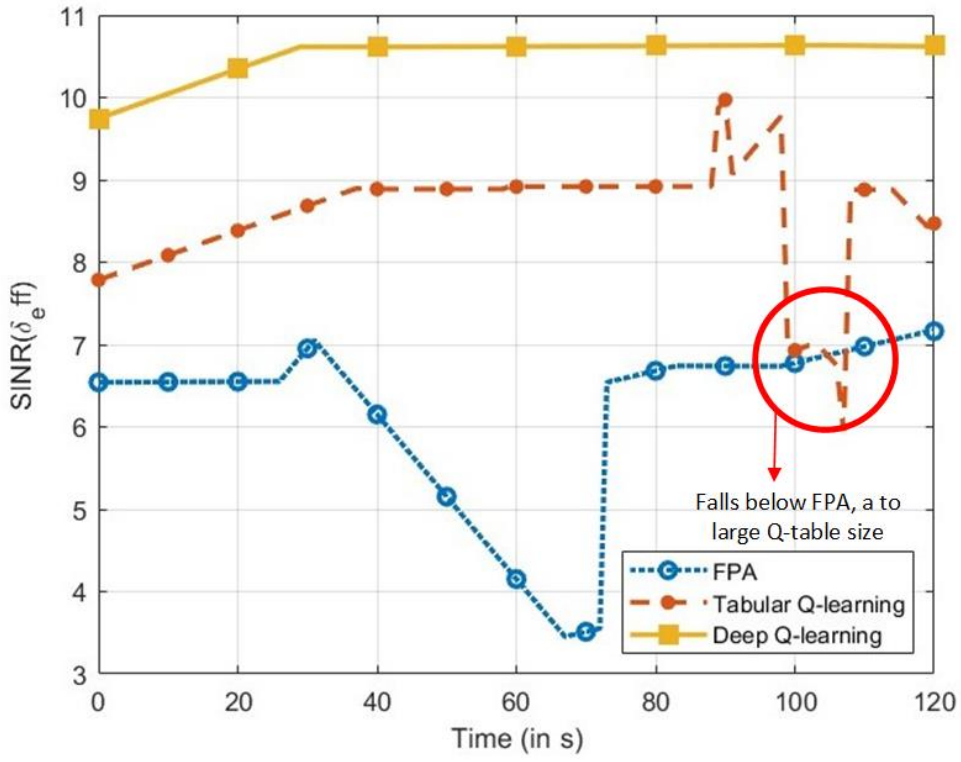

Figure 14. SINR vs. time @ 100 m.

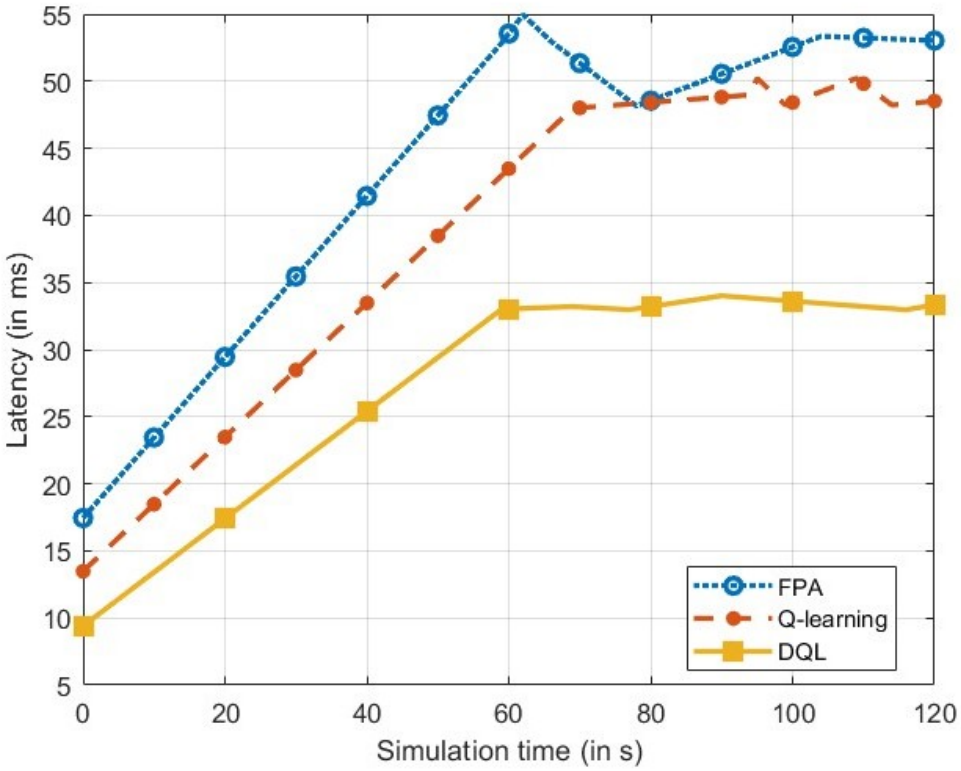

Figure 15. Latency at 100 m.

Finally, DQL performs the best with average, minimum, and maximum values of 10.51 dB, 9.74 dB, and 10.633 dB, respectively, which is indicative of steady performance with high SINR values. It also shows a really low latency value of 34.01 ms. Even though it shows a 12.58% decrease in SINR values and 32.6% increase in latency values as compared to 50 m, it is still the best-performing algorithm. It implements link adaptation absent from FPA and eliminates the need for Q-tables by the use of neural networks.

It is to be noted that the overall performance of the UAV at 100 m deteriorates as compared to 50 m. The main cause for this is the fact that, due to the higher altitude, the number of interfering elements, i.e., gNBs that UAV has LoS with, increases.

8.2.3. UAV @ 120 m

In this section, we discuss the results when the UAV is flying at a height of 120 m. This is the highest altitude under consideration; therefore, the number of interfering elements is the maximum amongst the considered scenarios.

Figure 16 shows SINR values for UAV @ 120 m. When no link adaptation is employed, i.e., when the FPA method is used, the values vary from very 1.34 dB to 5.36 dB. This represents an unstable link as the SINR values vary drastically. For example, at about the 31 s mark, there is a sharp drop to 1.45 dB from 5.05 dB for about 30 s. However, at the 73 s mark, it increases till 5.38 dB sharply in a 6 s duration before falling back to 41.38 ms by the end of 100 s, after which it increases again and does not stay constant. The periods of steady and stable values are lesser as compared to when the UAVs are at a lower height, for example, between 0 and 26 s, and between 67 s and 73 s. As mentioned above, this is primarily because the FPA algorithm cannot take into account the changing interference conditions and, at a higher altitude of 100 m, the increase in interfering elements leads to a change in the number of interfering elements. This also coincides with the latency values, which see delays of up to 67.447 ms, as shown in Figure 17.

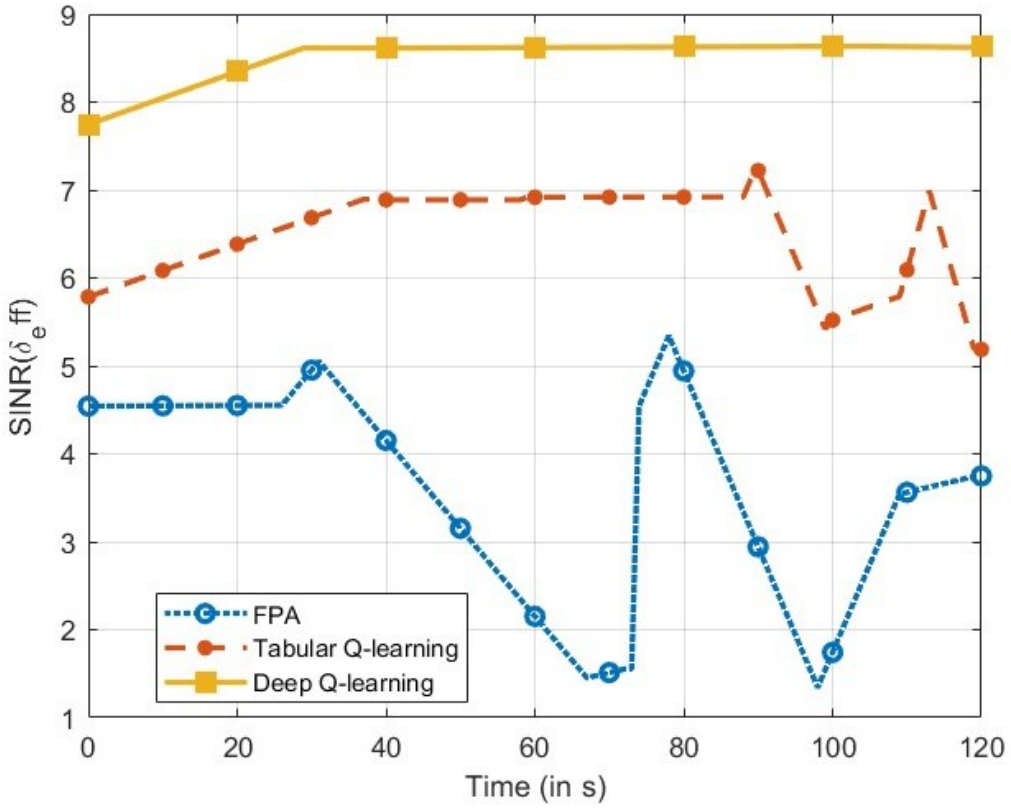

Figure 16. SINR vs. time @ 120 m.

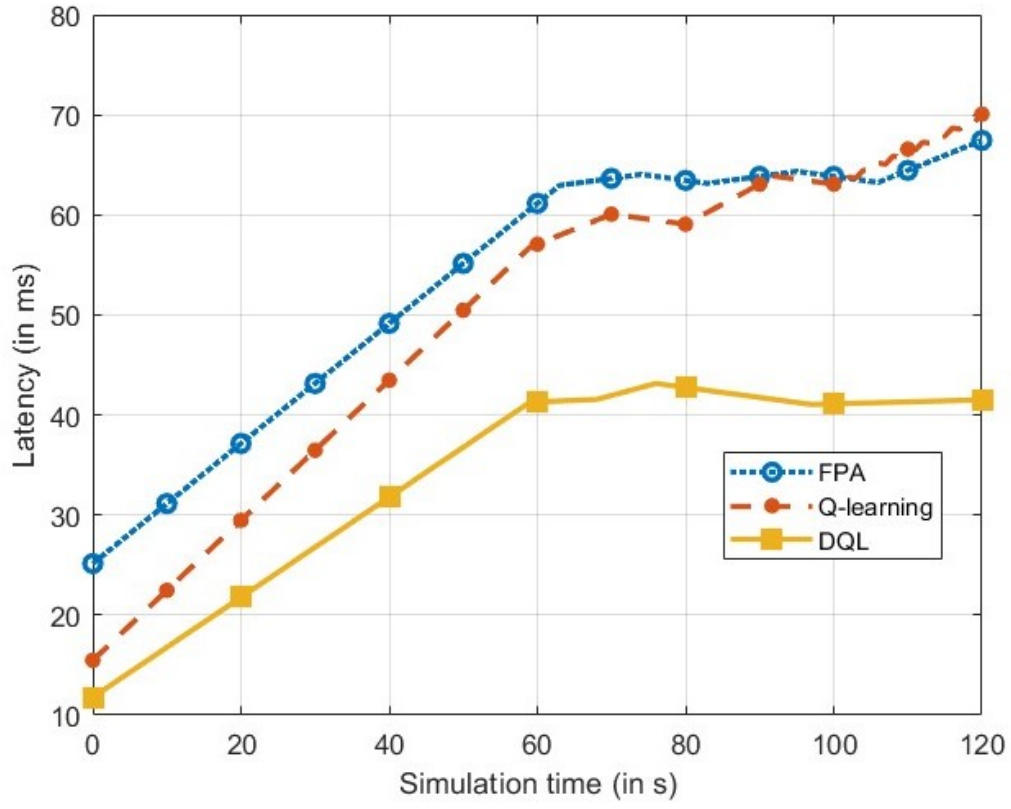

Figure 17. Latency at 120 m.

Q-learning starts with a lower value of 5.78 dB as compared to its 100 m performance. It maintains a steady performance value of close to 6.8 dB till about 88 s. This is indicative of good performance for the link; however, after about 89 s, it suffers a drastic dip in performance and undergoes a series of peaks and troughs till the end of simulation. When the simulation progresses, the Q-table becomes large and achieving the optimal solution becomes difficult. This is also reflected in the link latency, as shown in Figure 17.

Finally DQL performs the best with an average, minimum and maximum values of 8.51 dB, 7.72 dB and 8.6354 dB respectively, which is indicative of steady performance with high SINR values. It also shows a really low latency value of 43.15 ms. Even though it shows a 15.58% decrease in SINR values and 26.87% increase in latency values as compared to 100 m, it is still the best best performing algorithm. It implements link adaptation absent from FPA and eliminates the need for Q-tables by the use of neural networks.

It is to be noted that the overall performance of the UAV at 120 m is the worst as compared to 50 m and 100 m. The main cause for this is the fact that, due to the higher altitude, the number of interfering elements, i.e., gNBs that UAV has LoS with, increases.

Figure 18 shows the SINR values for three different speeds of the UAV. It is seen that, with increasing speeds of the UAV, it does not see any drastic changes in SINR values. The average δ value falls from about 9.5 dB at 10 m/s to 8.5 dB at 30 m/s. This is a clear indication that the DQL algorithm deals well with high speeds of the UAV. Table 8 summarises the results with different speeds.

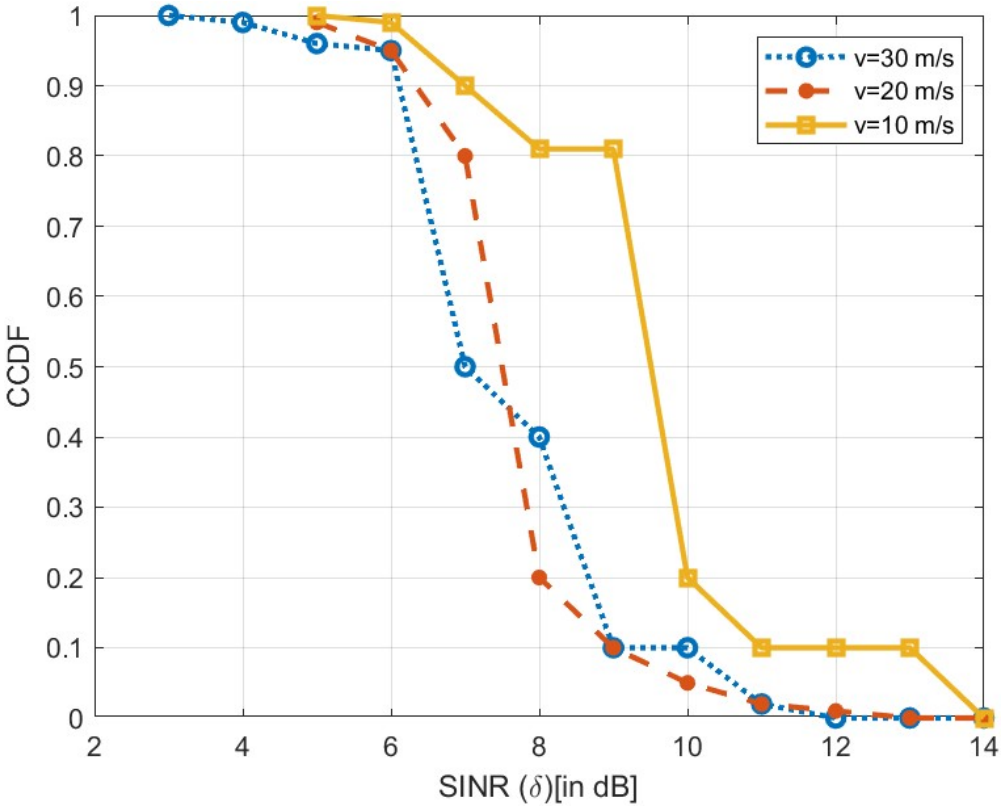

Figure 18. δ at different speeds of UAV.

Table 8. δ at different speeds of UAV.

UAV Speed	Average SINR
10 m/s	9.5 dB
20 m/s	9.5 dB
30 m/s	8.5 dB

Figure 19 shows the observed normalised runtime. Compared to FPA and tabular Q-learning, the proposed DQL algorithm has a shorter runtime and performs better in terms of convergence and capacity. As the no. of UAVs h increases, the runtime complexity of FPA and tabular Q-learning increase rapidly, but the runtime of DQL remains relatively low. At h = 4, the DQL algorithm only took 4% of the runtime of the other two algorithms. This behaviour is observed because the increase in h means that the number of interfering elements increases. FPA and tabular Q-learning are not able to accommodate these changing interfering conditions effectively, but the proposed DQL performs better than them.

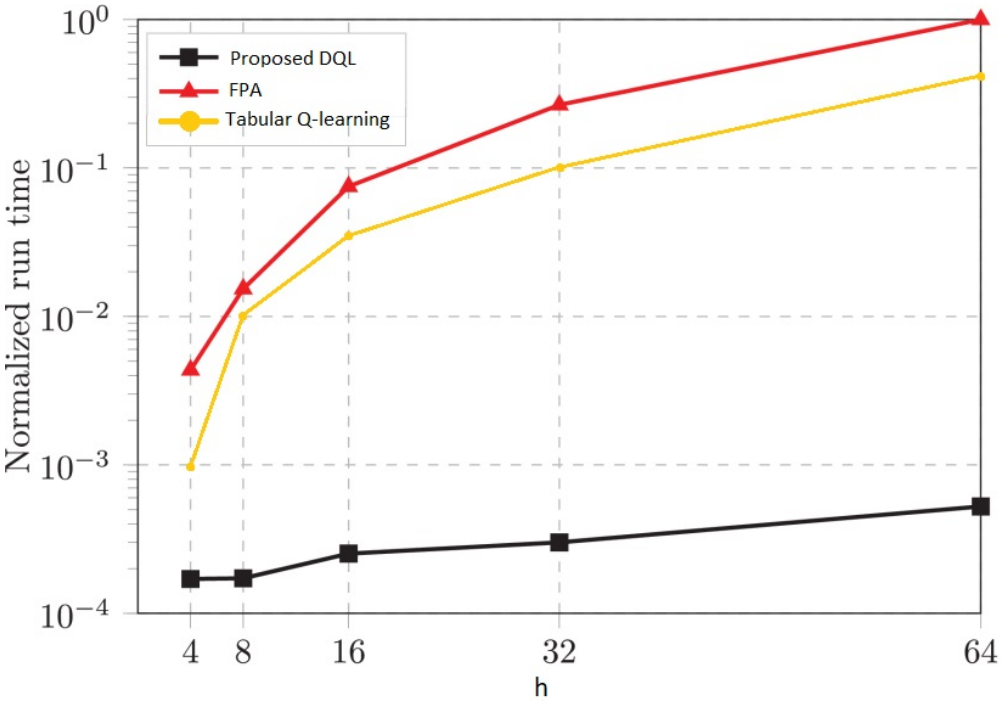

Figure 19. Normalised runtimes as a function of the no. of UAVs h.

Another way to visualise and evaluate an agent's performance during its training process is the reward plot. Figure 20 shows the reward plot for the proposed DQL method. As expected, the rewards are low in the beginning as the agent is learning to find the best solution, but they increase as the episodes progress, as the agent learns the best solution and keeps leveraging that knowledge as the proposed model employs an epsilon greedy policy.

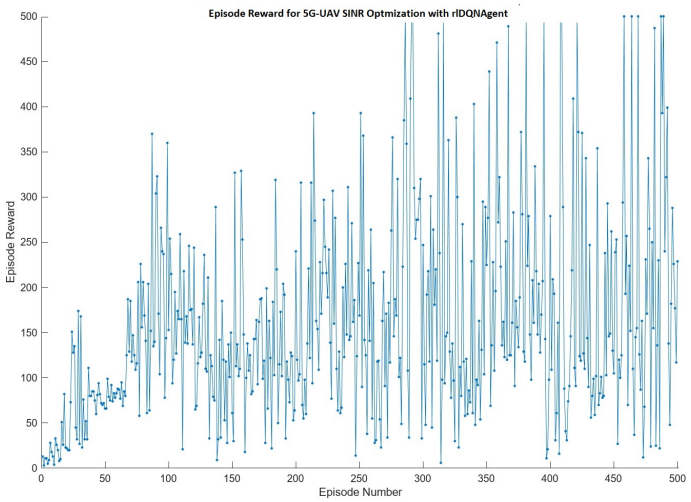

Figure 20. No. of episodes vs. rewards.

8.3. Outcomes

In this section, the outcomes of the performance measures are presented:

1. *Convergence:* We observe normalised convergence under (17), where δ_0^{thresh} = 5 dB. Each time step is equal to one radio sub-frame, with a duration of 1 ms. With an increase in the number of "h", the number of episodes necessary to achieve convergence also increases, with little to no impact on the threshold. δ_0^{thresh} since $h >> \delta_0^{thresh}$.
2. *Runtime:* The normalised runtime is observed. As h increases, the complexity of runtime complexity also increases for the proposed algorithm. This is justified because of an increased number of interfering elements.
3. *Coverage:* Coverage is defined by $1 - F_r(\delta)$, which evidently improves everywhere. Coverage improves where δ increases monotonically.

Additionally, the proposed solution is compared against some of the state-of-the-art methods implemented in the environment used in our work. The following Table 9 summarises the results. It is observed that the proposed DQL is considerably better than the other methods. SINR sees a 41.66% increase while using DQL as compared to mean-field game theory, the next best algorithm.

Table 9. Comparison of results with state-of-the-art method.

Reference	Algorithm	Average SINR Value
[29]	CNN	6 dB
[48]	Mean-field Q-Learning	0.30 dB
[49]	PSO-based Power Allocation	4 dB
[37]	Q-Learning	6.5 dB
[38]	Mean-Field Game Theory	6.42 dB
Proposed	Proposed	9.095 dB

In Figure 21, we plot the average SINR values of different methods as well as the proposed method and observe that the proposed method performs better than the existing methods in effectively mitigating interference. This can be explained by the following reasons: convolutional neural networks (CNNs) often require large memory spaces to store the weights and activations of the convolutional layers; this results in large overhead and complex design as opposed to the proposed DQL. Mean-field Q-learning has scalability limitations, unable to cope with a large number of agents and the consequent large growth of the action space. Particle swarm optimization can also be used for power allocation to mitigate interference. However, this method is designed to exploit the best solutions in the search spaces to a global optimum, which may not be the best trade-off between exploration–exploitation. Mean-field game theory for interference mitigation is also one of the methods amongst the existing works. However, this method is computationally complex, expensive, and challenging.

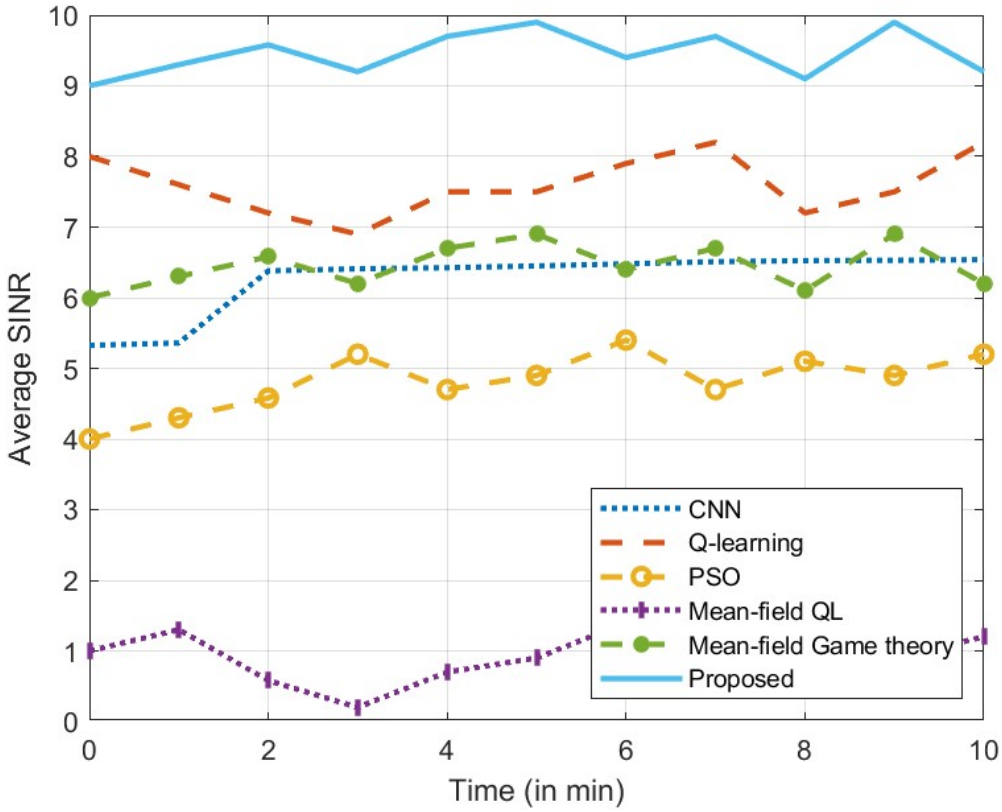

Figure 21. Comparison of results with state-of-the-art method.

9. Conclusions

The proposed algorithm maximises the SINR of UAVs in a 5G network by using DRL for power control and interference mitigation. It overcomes the unique challenges posed by the 3D motion of UAVs and the resulting air-to-ground interference by leveraging the cloud-based architecture of 5G systems. The algorithm outperforms traditional algorithms like Q-learning and FPA. The proposed DQL requires the UAV to send its position coordinates and receive the SINR at each step of the way to the gNB. However, it has no previous information about the CSI, which effectively means that channel estimates and associated training sequences are not needed. Thus, overall UAV feedback is reduced when the UAV sends its coordinates. Therefore, the interference mitigation proposed in this paper surpasses the existing methods, outperforming the next best algorithm by 41.66%, and offers a solution to one of the major obstacles in the implementation of 5G-connected UAVs, namely air-to-ground interference.

Author Contributions: Conceptualization, A.W.; methodology, A.W.; software, A.W.; validation, A.W. and S.A.-R.; formal analysis, A.W.; investigation, A.W.; resources, S.A.-R.; writing—original draft preparation, A.W. and S.A.-R.; writing—review and editing, A.W., S.A.-R. and G.I.; supervision, S.A.-R. and G.I.; funding acquisition, A.T. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Engineering and Physical Sciences Research Council (EPSRC) and Satellite Applications Catapult.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict 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. Federal Aviation Administration, Summary of Small Unmanned Aircraft Rule (Part 107). 2016. Available online: https://www.faa.gov/newsroom/small-unmanned-aircraft-systems-uas-regulations-part-107 (accessed on 12 September 2023).
2. Zeng, Y.; Lyu, J.; Zhang, R. Cellular-Connected UAV: Potential, Challenges, and Promising Technologies. *IEEE Wirel. Commun.* **2019**, *26*, 120–127. [CrossRef]
3. Khan, S.K.; Naseem, U.; Siraj, H.; Razzak, I.; Imran, M. The role of unmanned aerial vehicles and mmWave in 5G: Recent advances and challenges. *Trans. Emerg. Telecommun. Technol.* **2021**, *32*, e4241. . [CrossRef]
4. Jiang, X.; Sheng, M.; Zhao, N.; Xing, C.; Lu, W.; Wang, X. Green UAV communications for 6G: A survey. *Chin. J. Aeronaut.* **2022**, *35*, 19–34. .: 10.1016/j.cja.2021.04.025. [CrossRef]
5. Mozaffari, M.; Lin, X.; Hayes, S. Toward 6G with Connected Sky: UAVs and Beyond. *IEEE Commun. Mag.* **2021**, *59*, 74–80. [CrossRef]
6. Cicek, C.T.; Gultekin, H.; Tavli, B.; Yanikomeroglu, H. UAV Base Station Location Optimization for Next Generation Wireless Networks: Overview and Future Research Directions. In Proceedings of the 2019 1st International Conference on Unmanned Vehicle Systems-Oman (UAVS), Muscat, Oman, 5–7 February 2019; pp. 1–6. [CrossRef]
7. Yan, K.; Ma, L.; Zhang, Y. Research on the Application of 5G Technology in UAV Data Link. In Proceedings of the 2020 IEEE 9th Joint International Information Technology and Artificial Intelligence Conference (ITAIC), Chongqing, China, 11–13 December 2020; pp. 1115–1118. [CrossRef]
8. Gopalakrishnan, S.K.; Al-Rubaye, S.; Inalhan, G. Adaptive UAV Swarm Mission Planning by Temporal Difference Learning. In Proceedings of the 2021 IEEE/AIAA 40th Digital Avionics Systems Conference (DASC), San Antonio, TX, USA, 3–7 October 2021; pp. 1–10. [CrossRef]
9. Whitworth, H.; Al-Rubaye, S.; Tsourdos, A.; Jiggins, J.; Silverthorn, N.; Thomas, K. Aircraft to Operations Communication Analysis and Architecture for the Future Aviation Environment. In Proceedings of the 2021 IEEE/AIAA 40th Digital Avionics Systems Conference (DASC), San Antonio, TX, USA, 3–7 October 2021; pp. 1–8. [CrossRef]
10. Zeng, Y.; Guvenc, I.; Zhang, R.; Geraci, G.; Matolak, D. *UAV Communications for 5G and Beyond*; Wiley: Hoboken, NJ, USA, 2020.
11. Busari, S.A.; Huq, K.M.S.; Mumtaz, S.; Rodriguez, J.; Fang, Y.; Sicker, D.C.; Al-Rubaye, S.; Tsourdos, A. Generalized Hybrid Beamforming for Vehicular Connectivity Using THz Massive MIMO. *IEEE Trans. Veh. Technol.* **2019**, *68*, 8372–8383. [CrossRef]
12. Warrier, A.; Al-Rubaye, S.; Panagiotakopoulos, D.; Inalhan, G.; Tsourdos, A. Interference Mitigation for 5G-Connected UAV using Deep Q-Learning Framework. In Proceedings of the 2022 IEEE/AIAA 41st Digital Avionics Systems Conference (DASC), Portsmouth, VA, USA, 18–22 September 2022.
13. Li, C.; Sun, S.C.; Al-Rubaye, S.; Tsourdous, A.; Guo, W. Uncertainty Propagation in Neural Network Enabled Multi-Channel Optimisation. In Proceedings of the 2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring), Antwerp, Belgium, 25–28 May 2020; pp. 1–4. [CrossRef]
14. Vasudevan, S.; Pupala, R.N.; Sivanesan, K. Dynamic eICIC—A Proactive Strategy for Improving Spectral Efficiencies of Heterogeneous LTE Cellular Networks by Leveraging User Mobility and Traffic Dynamics. *IEEE Trans. Wirel. Commun.* **2013**, *12*, 4956–4969. [CrossRef]
15. Chowdary, A.; Chopra, G.; Kumar, A.; Cenkeramaddi, L.R. Enhanced User Grouping and Pairing Scheme for CoMP-NOMA-based Cellular Networks. In Proceedings of the 2022 14th International Conference on COMmunication Systems NETworkS (COMSNETS), Bangalore, India, 4–8 January 2022; pp. 319–323. [CrossRef]
16. Pan, J.; Ye, N.; Yu, H.; Hong, T.; Al-Rubaye, S.; Mumtaz, S.; Al-Dulaimi, A.; Chih-Lin, I. AI-Driven Blind Signature Classification for IoT Connectivity: A Deep Learning Approach. *IEEE Trans. Wirel. Commun.* **2022**, *21*, 6033–6047. [CrossRef]
17. Liu, L.; Zhang, S.; Zhang, R. Multi-Beam UAV Communication in Cellular Uplink: Cooperative Interference Cancellation and Sum-Rate Maximization. *IEEE Trans. Wirel. Commun.* **2019**, *18*, 4679–4691. [CrossRef]
18. Kosta, C.; Hunt, B.; Quddus, A.U.; Tafazolli, R. On interference avoidance through inter-cell interference coordination (ICIC) based on OFDMA mobile systems. *IEEE Commun. Surv. Tutor.* **2012**, *15*, 973–995. [CrossRef]
19. Amorim, R.; Nguyen, H.; Wigard, J.; Kovacs, I.Z.; Sorensen, T.B.; Biro, D.Z.; Sorensen, M.; Mogensen, P. Measured uplink interference caused by aerial vehicles in LTE cellular networks. *IEEE Wirel. Commun. Lett.* **2018**, *7*, 958–961. [CrossRef]
20. Chu, E.; Kim, J.M.; Jung, B.C. Interference Analysis of Directional UAV Networks: A Stochastic Geometry Approach. In Proceedings of the 2019 Eleventh International Conference on Ubiquitous and Future Networks (ICUFN), Zagreb, Croatia, 2–5 July 2019; pp. 9–12. [CrossRef]
21. Rahmati, A.; Hosseinalipour, S.; Yapici, Y.; He, X.; Guvenc, I.; Dai, H.; Bhuyan, A. Interference Avoidance in UAV-Assisted Networks: Joint 3D Trajectory Design and Power Allocation. In Proceedings of the 2019 IEEE Global Communications Conference (GLOBECOM), Waikoloa, HI, USA, 9–13 December 2019; pp. 1–6. [CrossRef]
22. Wu, Q.; Zeng, Y.; Zhang, R. Joint Trajectory and Communication Design for Multi-UAV Enabled Wireless Networks. *IEEE Trans. Wirel. Commun.* **2018**, *17*, 2109–2121. [CrossRef]
23. Shen, C.; Chang, T.H.; Gong, J.; Zeng, Y.; Zhang, R. Multi-UAV Interference Coordination via Joint Trajectory and Power Control. *IEEE Trans. Signal Process.* **2020**, *68*, 843–858. [CrossRef]
24. Pang, X.; Mei, W.; Zhao, N.; Zhang, R. Intelligent Reflecting Surface Assisted Interference Mitigation for Cellular-Connected UAV. *IEEE Wirel. Commun. Lett.* **2022**, *11*, 1708–1712. [CrossRef]

25. Ghavimi, F.; Jantti, R. Energy-Efficient UAV Communications with Interference Management: Deep Learning Framework. In Proceedings of the 2020 IEEE Wireless Communications and Networking Conference Workshops (WCNCW), Seoul, Republic of Korea, 6–9 April 2020; pp. 1–6. [CrossRef]
26. Letaief, K.B.; Chen, W.; Shi, Y.; Zhang, J.; Zhang, Y.J.A. The Roadmap to 6G: AI Empowered Wireless Networks. *IEEE Commun. Mag.* **2019**, *57*, 84–90. [CrossRef]
27. Lee, W.; Kim, M.; Cho, D.H. Deep power control: Transmit power control scheme based on convolutional neural network. *IEEE Commun. Lett.* **2018**, *22*, 1276–1279. [CrossRef]
28. Alkhateeb, A.; Alex, S.; Varkey, P.; Li, Y.; Qu, Q.; Tujkovic, D. Deep learning coordinated beamforming for highly-mobile millimeter wave systems. *IEEE Access* **2018**, *6*, 37328–37348. [CrossRef]
29. Luo, C.; Ji, J.; Wang, Q.; Yu, L. Online Power Control for 5G Wireless Communications: A Deep Q-Network Approach. In Proceedings of the 2018 IEEE International Conference on Communications (ICC), Kansas City, MO, USA, 20–24 May 2018; pp. 1–6. [CrossRef]
30. Jang, H.S.; Lee, H.; Quek, T.Q.S. Deep Learning-Based Power Control for Non-Orthogonal Random Access. *IEEE Commun. Lett.* **2019**, *23*, 2004–2007. [CrossRef]
31. Mismar, F.B.; Evans, B.L.; Alkhateeb, A. Deep Reinforcement Learning for 5G Networks: Joint Beamforming, Power Control, and Interference Coordination. *IEEE Trans. Commun.* **2020**, *68*, 1581–1592. [CrossRef]
32. Wang, S.; Liu, H.; Gomes, P.H.; Krishnamachari, B. Deep reinforcement learning for dynamic multichannel access in wireless networks. *IEEE Trans. Cogn. Commun. Netw.* **2018**, *4*, 257–265. [CrossRef]
33. Zhou, P.; Fang, X.; Wang, X.; Long, Y.; He, R.; Han, X. Deep learning-based beam management and interference coordination in dense mmWave networks. *IEEE Trans. Veh. Technol.* **2018**, *68*, 592–603. [CrossRef]
34. Xia, W.; Zheng, G.; Zhu, Y.; Zhang, J.; Wang, J.; Petropulu, A.P. A deep learning framework for optimization of MISO downlink beamforming. *IEEE Trans. Commun.* **2019**, *68*, 1866–1880. [CrossRef]
35. Park, K.; Lee, J.; Ryu, H.; Kim, Y. A Novel Cell Deployment for UAM Communications in 5G-Advanced Network. In Proceedings of the 2022 IEEE Globecom Workshops (GC Wkshps), Rio de Janeiro, Brazil, 4–8 December 2022; pp. 1431–1436. [CrossRef]
36. Zhu, L.; Zhang, J.; Xiao, Z.; Cao, X.; Wu, D.O.; Xia, X.G. Joint Power Control and Beamforming for Uplink Non-Orthogonal Multiple Access in 5G Millimeter-Wave Communications. *IEEE Trans. Wirel. Commun.* **2018**, *17*, 6177–6189. [CrossRef]
37. Chiang, H.L.; Chen, K.C.; Rave, W.; Marandi, M.K.; Fettweis, G. Multi-UAV mmWave Beam Tracking using Q-Learning and Interference Mitigation. In Proceedings of the 2020 IEEE International Conference on Communications Workshops (ICC Workshops), Dublin, Ireland, 7–11 June 2020; pp. 1–7. [CrossRef]
38. Zhang, Z.; Li, L.; Liang, W.; Li, X.; Gao, A.; Chen, W.; Han, Z. Downlink Interference Management in Dense Drone Small Cells Networks Using Mean-Field Game Theory. In Proceedings of the 2018 10th International Conference on Wireless Communications and Signal Processing (WCSP), Hangzhou, China, 18–20 October 2018; pp. 1–6. [CrossRef]
39. Sharma, M.K.; Zappone, A.; Debbah, M.; Assaad, M. Deep Learning Based Online Power Control for Large Energy Harvesting Networks. In Proceedings of the ICASSP 2019—2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Brighton, UK, 12–17 May 2019; pp. 8429–8433. [CrossRef]
40. Institution (ETSI), E.T.S. System Architecture for the 5G System-3GPP TS 23.501 v16 Rel16. 2020. Available online: https://www.etsi.org/deliver/etsi_ts/123500_123599/123501/16.06.00_60/ts_123501v160600p.pdf (accessed on 12 September 2023).
41. Kekki, S.; Featherstone, W.; Fang, Y.; Kuure, P.; Li, A.; Ranjan, A.; Purkayastha, D.; Jiangping, F.; Frydman, D.; Verin, G.; et al. ETSI White Paper No. 28. 2018, pp. 1–28. MEC in 5G networks. ETSI White Paper No. 28. 2018, pp. 1–28. Available online: https://www.etsi.org/images/files/ETSIWhitePapers/etsi_wp28_mec_in_5G_FINAL.pdf (accessed on 12 September 2023).
42. Lecun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436–444. [CrossRef]
43. Hornik, K. Approximation capabilities of multilayer feedforward networks. *Neural Netw.* **1991**, *4*, 251–257. [CrossRef]
44. Do, D.T.; Le, C.B. Impact of fixed power allocation in wireless energy harvesting NOMA networks. *Int. J. Commun. Syst.* **2019**, *32*, e4016. [CrossRef]
45. Zhu, M.; Wang, C.X.; Hua, B.; Kai, M.; Jiang, S.; Yao, M. *3GPP TR 38.901 Channel Model*; John Wiley & Sons, Inc.: Hoboken, NJ, USA, 2021; pp. 1–35. [CrossRef]
46. Evolved Universal Terrestrial Radio Access (E-UTRA); Physical Layer Procedures; ETSI TS: 2017; Available online: https://www.etsi.org/deliver/etsi_ts/136200_136299/136213/14.02.00_60/ts_136213v140200p.pdf (accessed on 12 September 2023).
47. Huang, G.B. Learning capability and storage capacity of two-hidden-layer feedforward networks. *IEEE Trans. Neural Netw.* **2003**, *14*, 274–281. [CrossRef] [PubMed]
48. Sun, Y.; Li, L.; Cheng, Q.; Wang, D.; Liang, W.; Li, X.; Han, Z. Joint Trajectory and Power Optimization in Multi-Type UAVs Network with Mean Field Q-Learning. In Proceedings of the 2020 IEEE International Conference on Communications Workshops (ICC Workshops), Dublin, Ireland, 7–11 June 2020; pp. 1–6. [CrossRef]
49. Liu, W.; Niu, G.; Cao, Q.; Pun, M.O.; Chen, J. Particle Swarm Optimization for Interference-Limited Unmanned Aerial Vehicle-Assisted Networks. *IEEE Access* **2020**, *8*, 174342–174352. [CrossRef]