Double Deep Reinforcement Learning–Based UAV Positioning for Throughput Optimization in Wireless Networks

Berke Kılınç*†, Mehmet Önder Efe*
* Department of Computer Engineering, Hacettepe University
† Communication and Information Technologies Division, ASELSAN Inc.
Ankara, Türkiye
berkekilinc@aselsan.com, onderefe@gmail.com

***Abstract*—This work investigates a reinforcement learning-based control framework for the autonomous movement and coordination of multiple Unmanned Aerial Vehicles (UAVs) in a wireless communication environment. The considered system includes UAVs performing sensing and relaying tasks, where mobility decisions directly affect the overall network performance. The main objective is to improve the communication quality of ground users by maximizing aggregate network throughput. To achieve this objective, a Double Deep Q-Network (DDQN) architecture is employed, where each UAV is assigned an individual learning agent. The agents learn role-specific movement policies while coordinating through interactions with the shared environment. Learning performance is further improved by using adaptive scaling and a custom reward function designed to capture variations in network utility. Simulation results show that the proposed approach outperforms baseline movement strategies in terms of utility. In addition, different task configurations, agent behaviors, and hyperparameter selections are examined to improve convergence speed and training stability. Overall, the results indicate that reinforcement learning is a promising method for cooperative UAV positioning in dynamic and interference-sensitive wireless communication scenarios.**

***Index Terms*—Unmanned Aerial Vehicles, Positioning, Deep-learning, DDQN, Communication, Throughput, Optimization**

I. Introduction

Unmanned Aerial Vehicles (UAVs) have gained considerable attention in recent years due to their versatility and effectiveness in applications such as wireless communication enhancement, data collection, and network support [1]. Modern communication systems increasingly face challenges arising from dynamic environmental conditions, fluctuating user demands, and strict performance requirements. Conventional fixed infrastructures often struggle to respond efficiently to these variations, creating a need for more flexible and adaptive solutions. In this context, UAV-assisted communication systems offer a compelling alternative by leveraging mobility and rapid reconfigurability to improve coverage, capacity, and service quality [2].

The inherent mobility of UAVs allows communication networks to dynamically adjust to changing scenarios, such as user movement or uneven traffic distribution. However, efficiently managing multiple UAVs operating simultaneously introduces significant complexity. The interaction between UAVs, users, and the wireless environment forms a highly dynamic multi-agent system, where traditional optimization and rule-based methods quickly become impractical or suboptimal [3]. As a result, intelligent decision-making mechanisms capable of learning and adapting in real time are required.

Machine learning–based approaches, particularly reinforcement learning (RL), have become effective methods for addressing adaptive decision-making problems in complex wireless environments. By enabling agents to learn optimal behaviors through interaction with the environment, RL techniques are well suited for decentralized and adaptive control problems. Among these methods, deep reinforcement learning algorithms such as Deep Q-Networks (DQN) have shown strong potential in handling high-dimensional state spaces and making sequential decisions in wireless networking applications [4], [5].

In this study, a DDQN-based positioning strategy is proposed for a multi-UAV system that operates under two main functions: communication relay and data fusion. The main objective is to maximize the overall system utility by enabling each UAV to learn its optimal position over time. The proposed method provides an adaptable architecture by achieving high overall system utility value by taking into account factors such as users' locations and movement restrictions.

The main contributions of this paper are summarized as follows:

- A cooperative UAV positioning framework is developed for a wireless communication scenario in which relay and sensing functions are jointly considered.
- A hybrid reward function is designed by combining relay-oriented and sensing-oriented utility values, enabling the UAV agents to improve the overall system throughput rather than focusing on a single communication objective.
- A DDQN-based learning architecture is employed within a decentralized execution structure, where each UAV independently selects its movement action using its trained policy during operation.

- The distributed agent-based structure provides a flexible simulation basis for extending the framework to larger cooperative UAV networks with multiple UAVs and different operational roles.

The remainder of this paper is organized as follows: the next section presents relevant background and related work. The proposed methodology is then detailed, followed by an experimental setup, evaluation results, and comparative analysis. Finally, the paper concludes with key findings and suggestions for future research.

II. Related Works

In recent years, the use of unmanned aerial vehicles in communication systems has become quite widespread. This is due to their ability to provide fast, flexible, and low-cost solutions in civil and military applications. UAV-assisted communication has found use cases in disaster recovery, remote sensing, temporary hotspot coverage, and battlefield networking [6] [7].

There are articles in the literature that focus on optimization-based UAV positioning that do not use machine learning. For example, in [8], an architecture was created to achieve maximum coverage using geometrical and coverage-based algorithms. In addition, reference [9] implemented UAV positioning using traditional methods to increase total data transmission in the environment.

With the increasing complexity of real-world scenarios and the need for real-time adaptation, learning-based methods, particularly reinforcement learning (RL) [10], have emerged as effective alternatives to optimize UAV placement [11]. There are many articles in the literature that have been developed using Q-Learning. In [12], optimization of the location of the UAV was obtained to maximize the value of the mean opinion score using Q-Learning. In [13], the energy efficiency for UAVs was maximized with the help of Q-Learning. In article [14], a connection between users was established using UAV in a disaster scenario. Q-learning was used to increase the quality of communication between users [15]. By creating a Q-Learning model, the sum utility value was maximized by changing the UAV locations for the primary network and secondary networks [16].

As the scope and number of possible situations in the studies increased, Deep Q-Learning based algorithms were studied instead of standard Q-Learning algorithms. With the use of the Deep Q-Learning algorithm, it is possible to work on very large size situation-action pairs [17]. In [18], the authors determined the route of the UAV with the help of Deep Reinforcement Learning to ensure effectiveness in the data collection task. In [19], the authors maximized the coverage area using UAV without considering the channel effects in the calculation. In addition, the reward function was maximized using a distributed learning based Deep Q-Learning algorithm to increase the data transmission rate [20]. In [21], the authors used a deep learning based architecture for resource allocation and user association. In [22], a Deep Reinforcement Learning based architecture was created that maximizes the coverage area in the cellular network and minimizes the interference levels in the environment. In most of these studies, an algorithm was developed that tries to maximize the coverage area by neglecting the real-world channel conditions.

This work presents a unique decentralized DQN-based UAV control framework that employs a hybrid reward function, optimizes UAV location in both sensing and relay missions, and computes the channel status between UAV-user using actual channel state information (CSI). The scope of conventional DQN implementations is expanded by this hybrid task formulation. Although many existing studies in the literature develop a centralized algorithm for a single UAV, this study adopts a distributed architecture for two different UAVs.

The novelty of this work lies in the development of a DDQN-based cooperative UAV positioning framework that jointly considers relay and sensing functions through a hybrid task-aware reward structure. By combining relay-oriented and sensing-oriented utility values, the proposed model enables UAV agents to learn movement policies that improve the overall communication performance of the system. In addition, the distributed agent-based structure allows each UAV to make role-specific movement decisions in a shared environment, providing a flexible simulation basis for future extensions to larger cooperative UAV networks.

III. Methodology

This study proposes a Double Deep Q-Network (DDQN)-based framework for optimizing the positions of two UAVs in a wireless communication system. Each UAV independently makes the most appropriate location decision to optimize the total throughput of the primary and secondary networks and moves in two dimensions accordingly. The proposed architecture uses a Double Deep Q-Learning based architecture to find the most appropriate location.

An example scenario environment is shown in Fig. 1. In the proposed scenario, the ground transmitter and ground receiver that communicate with each other represent the primary network, while the source and emergency center (HAP) provide this information to be received from the environment via the secondary network and transferred to the relevant central control unit (HAP). The information received is processed in the central control unit and the most appropriate location of each UAV is decided.

In this architecture, UAV 1 maximizes the total throughput value of the primary network, while UAV 2 maximizes the total throughput value of the secondary network. The decision mechanism operates depending on the location of these units and the environmental conditions.

In this scenario, the transmitter, receiver, HAP, and source are located in different fixed positions, and the UAVs are adaptively repositioned to optimize communication and sensing performance relative to these ground nodes. Channel state information is continuously updated according to the UAV positions using the channel modeling approach in [16].

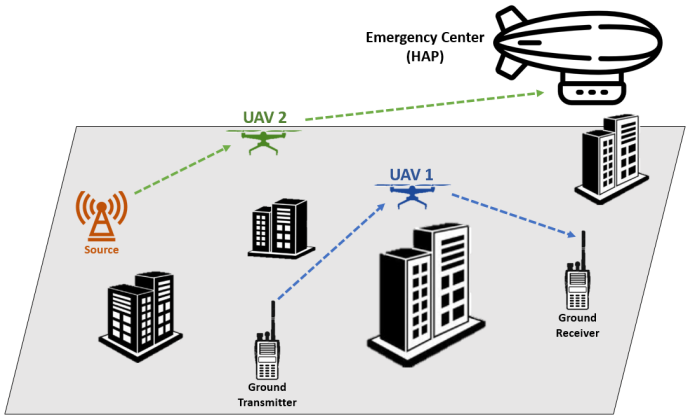

Fig. 1. System overview of the proposed DDQN-based UAV coordination architecture.

A. Q-learning

Q-Learning is a reinforcement learning algorithm that allows the model to learn an action policy that will take the highest expected rewards under certain environmental conditions. The Q value in this algorithm represents the cumulative reward value expected by the model for taking a particular action while in a particular state.

The Q-learning update rule is defined as follows:

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha\big[r_{t+1} + \gamma \max_{a'} Q(s_{t+1}, a') - Q(s_t, a_t)\big] \quad (1)$$

In this expression, s_t denotes the current state at time t, a_t represents the action taken at time t, r_{t+1} is the immediate reward received after executing action a_t, α is the learning rate, and γ denotes the discount factor that determines the importance of future rewards.

In Q-learning, the model's state, action and q-values are stored in a table called Q-table and the agent uses this Q-table as a look-up table to make the most appropriate decision. Thus, it makes the best action decision to be taken in any situation.

B. Double Deep Q-Learning Architecture

Deep Q-Learning is the decision process made using a neural network, which is used in the standard Q-Learning algorithm using a Q-table. In this way, adaptive learning is provided even in scenarios that contain multiple state information. Double Deep Q-Learning is an extension of the Deep Q-Learning algorithm and is used to solve the problem of overestimation bias [23].

In the standard DQN framework, the use of the max operator in the computation of target values often results in the selection of overestimated Q-values, which can create suboptimal policies. Double DQN mitigates this issue by separating the action selection and evaluation processes through the utilization of two distinct networks: an online (primary) network and a target network.

The target for the Q-value update in Double DQN is computed as:

$$y_t^{\text{DoubleDQN}} = r_t + \gamma Q_{\text{target}}(s_{t+1}, \arg\max_a Q_{\text{online}}(s_{t+1}, a; \theta_t); \theta_t^-) \quad (2)$$

Here, θ_t and θ_t^- represent the parameters of the online network and the target network, respectively. Unlike standard DQN where both selection and evaluation are performed using the target network, Double DQN selects the action with the online network and evaluates its value with the target network.

The training process involves periodically updating the target network's weights to match the online network, i.e., $\theta_t^- \leftarrow \theta_t$ every C steps.

Algorithm 1 Double Deep Q-Learning

```
Initialize online network Q(s, a; θ) with random weights θ
Initialize target network Q_target(s, a; θ⁻) with weights θ⁻ = θ
Initialize experience replay buffer D
for each episode do
    Initialize environment and get initial state s_0
    for each step t do
        Select action a_t using ε-greedy policy from Q(s_t, a; θ)
        Execute action a_t, observe reward r_t and next state s_{t+1}
        Store (s_t, a_t, r_t, s_{t+1}) in D
        Sample random mini-batch from D
        for each (s_j, a_j, r_j, s_{j+1}) in batch do
            a* ← arg max_a Q(s_{j+1}, a; θ)
            y_j ← r_j + γ Q_target(s_{j+1}, a*; θ⁻)
        end for
        Perform gradient descent on (y_j − Q(s_j, a_j; θ))²
        if step t mod C = 0 then
            Update target network: θ⁻ ← θ
        end if
    end for
end for
```

Fig. 2 shows the general architecture of the DDQN solution method proposed in this article. The information received from the environment is stored in the replay buffer as a state experience. This information is used to train the evaluation (online) and target networks in minibatches, and model training is performed over two different networks, thus obtaining more stable results.

C. Overview of the DDQN-Based UAV Model

In this study, each UAV is equipped with its own Double Deep Q-Network (DDQN) architecture. The state input for each UAV is represented in a simplified integer format as follows:

$$\text{State} = [10 \times y_{\text{uav}_1} + x_{\text{uav}_1},\ 10 \times y_{\text{uav}_2} + x_{\text{uav}_2}] \quad (3)$$

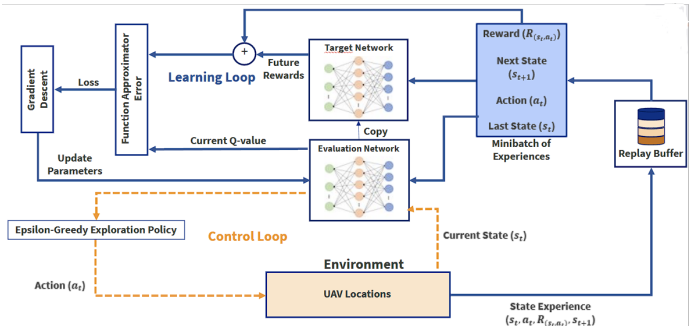

Fig. 2. Overall DDQN architecture.

This coding takes the location of both UAVs as state information and uses it to decide on the most appropriate action in the model, where x and y denote the coordinates of each UAV.

Each UAV uses its state information to decide on one of five different actions at each time step: moving forward, moving backward, moving right, moving left, and staying stationary.

The reward function is defined as a linear combination of the primary and secondary utilities:

$$R = \alpha \cdot U_{\text{primary}} + \beta \cdot U_{\text{secondary}} \tag{4}$$

Here, α and β are weighting coefficients that control the contribution of U_{primary} and $U_{\text{secondary}}$ to the hybrid reward function, respectively. Equal coefficient values were used in this study because both the primary and secondary network throughput values are intended to be improved without prioritizing one network over the other. The coefficient selection process and its impact on the final sum utility are presented in detail in the Performance Evaluation section.

In the proposed architecture, UAV 1 is assigned as the relay UAV, while UAV 2 is assigned as the sensing UAV. The throughput value of the system in which UAV 1 served as a relay was called the primary network utility, and the throughput value of the system in which UAV 2 served as a sensing device was called the secondary network throughput. In this way, they were able to obtain optimum positions specific to each task.

A hybrid reward function is designed to jointly consider the primary and secondary network throughput values. Each UAV is equipped with its own DDQN agent, enabling role-specific policy learning in a shared and cooperative multi-agent environment. The state observed by each agent includes the positions of both UAVs, allowing the learning process to capture how their relative locations affect interference patterns, spectrum awareness, and overall communication quality. Although the UAVs operate with different roles, one acting as a relay and the other performing sensing, their movements influence the same underlying system dynamics. Therefore, the relay and sensing throughput metrics are combined in a weighted manner to provide a shared reward signal that supports adaptive and cooperative UAV positioning.

The contribution of this work lies in the development of a decentralized DDQN-based UAV control framework that jointly optimizes positioning for sensing and relay missions under realistic wireless communication constraints. By utilizing a hybrid reward structure and considering real-time channel state information, the proposed method outperforms classical Q-learning approaches, enabling coordinated control of multiple UAVs in different missions. These improvements produce a decision-making framework that is more performance-efficient, flexible, and realistic for practical implementations.

IV. Performance Evaluation

The DDQN architecture proposed in this article was established in Python with the help of Tensorflow and Keras libraries. A fully connected feed-forward network with two hidden layers was used for stable Q-value estimation. ReLU was employed as the activation function, and the network was trained using the Adam optimizer with a learning rate of 1×10^{-4}. Experience replay method was used to increase the model learning stability and previous experiences were used for training with minibatch. Detailed information about the hyperparameters is given in Table I.

In the proposed hybrid reward function, the coefficients α and β determine the relative contribution of the primary and secondary network utilities. In this study, equal coefficient values were used because the objective is to improve the throughput of both networks without prioritizing one over the other. If one network is considered more critical in a different mission scenario, these coefficients can be adjusted accordingly to assign higher importance to either the relay-oriented or sensing-oriented objective. To justify the selected coefficient values, different equal scaling factors were tested for α and β. Table II shows the final sum utility values obtained under different coefficient settings.

As shown in Table II, small coefficient values resulted in lower final sum utility because the reward variations were less distinguishable during training. Increasing the coefficient values improved the learning sensitivity of the DDQN agents. However, further increasing the coefficients reduced the final

TABLE I
Parameters of DDQN Architecture

Hyperparameter	Value
Number of layers	2 [128, 64 neurons]
Optimizer	Adam
Activation function	ReLU
Target network update frequency	50 steps
Experience replay buffer size	50000
Batch size	10
Initial and final exploration coefficient	ε_i = 0.95, ε_f = 0.05
Training iteration count	300
Steps per iteration	1000
Learning rate	0.0001
Discount factor	0.80

TABLE II
EFFECT OF REWARD COEFFICIENT SELECTION ON FINAL SUM UTILITY

α	β	**Final Sum Utility**
10	10	0.0382
100	100	0.0423
500	500	0.0441
1000	1000	**0.0476**
1500	1500	0.0435
2000	2000	0.0428

utility, indicating that excessive reward scaling may negatively affect learning stability. Therefore, $\alpha = \beta = 1000$ was selected for the remaining simulations.

Model training was done for 300 iterations and 1000 steps, and the UAVs were reset to their initial positions in each iteration and moved as many steps as the channel state information between them and the users on the ground. The output obtained for model training is as shown in Fig. 3. This figure gives the total throughput value of the primary and secondary networks for 300 iterations.

Fig. 4 gives the average sum utility values of model training at different step numbers for each iteration. Within the scope of this study, 5 different step values were tested for 300 iterations and performance results were obtained. As can be seen from the figure, model performance was further improved as the number of steps increased. As the number of steps increased, the fluctuations in the learning process of the model decreased and learning became more stable.

An example image of a trained model in the test environment is as shown in Fig. 5. Here, the cyan colored UAV is the UAV that acts as a relay for the primary network, and the magenta colored UAV is the UAV that acts as a sensing device for the secondary network. As can be seen from the figure, the UAVs have moved from the initial position to the most suitable point to increase the system throughput.

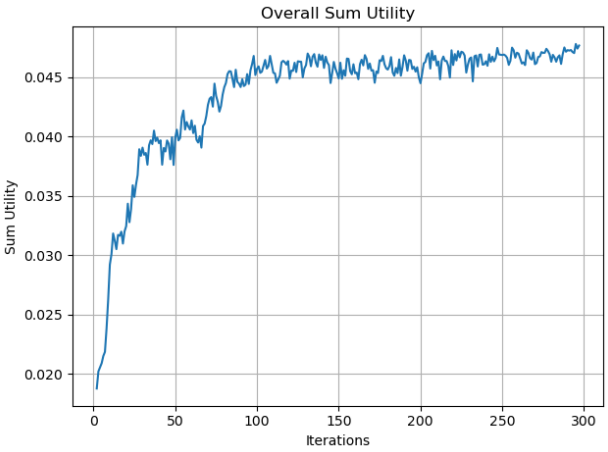

Fig. 3. Variation of sum utility with respect to number of iterations for training.

It was observed that the obtained results showed better

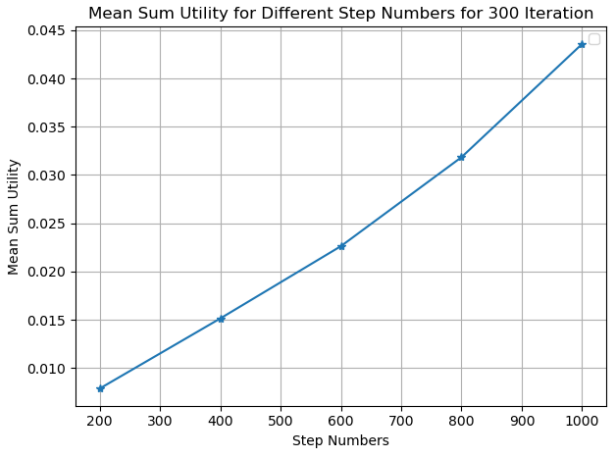

Fig. 4. Variation of overall sum utility for different step numbers.

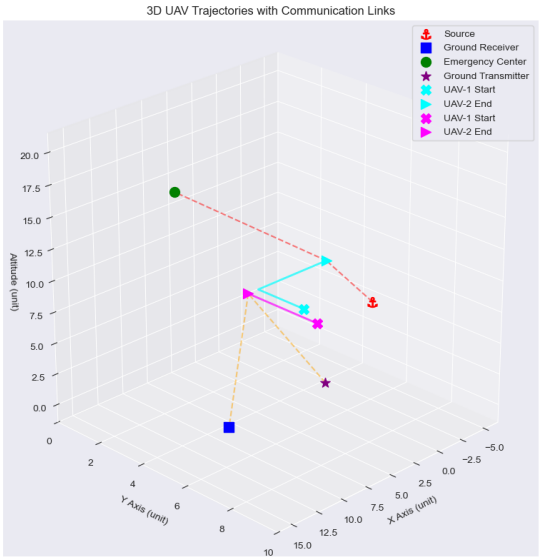

Fig. 5. Illustration of the UAV flight trajectories.

performance than the results found in the literature. In [16], using standard Q-learning, a total throughput of 0.37 was obtained in 200 iterations with 30000 steps in each iteration. This total throughput value is the overall throughput value for 30000 steps. As a comparison with the original method, a total throughput value of 0.0476 was obtained in 1000 steps, and when calculated for 30000 steps, it was observed that the performance increase was almost 3.86 times higher than the classical Q-learning algorithm. These results demonstrate that the DDQN structure and utility-aware training significantly improve the system's adaptability and mission effectiveness.

The work in [16] is selected as the main quantitative baseline because it is the most closely related study in terms of scenario definition and system objective. Similar to the proposed framework, it considers a mission-critical communication environment supported by UAV with primary and secondary network utilities and relay/sensing-related func-

tions. Therefore, a numerical comparison with this baseline is conducted in order to evaluate the proposed DDQN-based method under a scenario-consistent reference.

Beyond the main quantitative baseline, other RL- and DRL-based UAV communication studies were also examined. However, no additional study was identified that uses the same system environment, relay/sensing task structure, optimization objective, and sum-utility-based performance metric as the proposed framework. Therefore, a direct numerical comparison with these studies was not conducted. Instead, [24] and [20] are considered as supporting references, since they demonstrate the relevance of utility-oriented optimization and learning-based decision-making in UAV-assisted communication systems.

V. Conclusion

In this paper, we proposed a Double Deep Q-Learning (DDQN)-based architecture for optimizing the positioning and task assignment of UAVs in mission-critical communication environments. Instead of relying on standard Q-learning algorithms commonly found in the literature, a DDQN-based architecture was employed, resulting in improved adaptability in large state-action spaces, better convergence behavior, and more accurate decision-making. Compared with the closest scenario-consistent Q-learning-based baseline, the proposed framework achieved approximately 3.86 times higher step-normalized utility in the considered simulation setting.

Future work will extend the proposed framework in three main directions. First, the number of UAVs and ground users will be increased to evaluate scalability. Second, user mobility and time-varying traffic demand will be included to create a more dynamic environment. Finally, different channel conditions will be tested to assess robustness.

References

[1] N. C. Luong, D. T. Hoang, S. Gong, D. Niyato, P. Wang, Y.-C. Liang, and D. I. Kim, "Applications of deep reinforcement learning in communications and networking: A survey," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 4, pp. 3133–3174, 2019.

[2] M. M. Azari, F. Rosas, A. Chiumento, and S. Pollin, "Ultra reliable UAV communication using altitude and cooperation diversity," *IEEE Transactions on Communications*, vol. 66, no. 1, pp. 330–344, 2018.

[3] B. Li, Z. Fei, and Y. Zhang, "UAV communications for 5G and beyond: Recent advances and future trends," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 2241–2263, 2019.

[4] F. Cheng, S. Zhang, and W. Zhuang, "Reinforcement learning for delay-aware mobile edge computing in wireless networks," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 3, pp. 2773–2786, 2018.

[5] H. Wang, Z. Wang, and J. Chen, "Deep reinforcement learning for dynamic spectrum access in wireless networks," *IEEE Transactions on Cognitive Communications and Networking*, vol. 6, no. 3, pp. 1046–1059, 2020.

[6] A. Fotouhi, H. Qiang, M. Ding, M. Hassan, L. G. Giordano, A. Garcia-Rodriguez, and J. Yuan, "Survey on UAV cellular communications: Practical aspects, standardization advancements, regulation, and security challenges," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 4, pp. 3417–3442, 2019. https://doi.org/10.1109/COMST.2019.2909625

[7] L. Gupta, R. Jain, and G. Vaszkun, "Survey of important issues in UAV communication networks," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 2, pp. 1123–1152, 2016. https://doi.org/10.1109/COMST.2015.2495297

[8] M. Mozaffari, W. Saad, M. Bennis, and M. Debbah, "Mobile unmanned aerial vehicles (UAVs) for energy-efficient Internet of Things communications," *IEEE Transactions on Wireless Communications*, vol. 16, no. 11, pp. 7574–7589, Nov. 2017.

[9] Y. Zeng, R. Zhang, and T. J. Lim, "Wireless communications with unmanned aerial vehicles: Opportunities and challenges," *IEEE Communications Magazine*, vol. 54, no. 5, pp. 36–42, 2016. https://doi.org/10.1109/MCOM.2016.7470933

[10] C. J. C. H. Watkins and P. Dayan, "Q-learning," *Machine Learning*, vol. 8, pp. 279-292, 1992. https://doi.org/10.1007/BF00992698

[11] Y. Bai, H. Zhao, X. Zhang, Z. Chang, R. Jäntti, and K. Yang, "Toward autonomous multi-UAV wireless network: A survey of reinforcement learning-based approaches," *IEEE Communications Surveys & Tutorials*, vol. 25, no. 4, pp. 3038–3067, 2023.

[12] X. Liu, Y. Liu, and Y. Chen, "Reinforcement learning in multiple-UAV networks: Deployment and movement design," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 8, pp. 8036-8049, Aug. 2019. https://doi.org/10.1109/TVT.2019.2922849

[13] S. Lee, H. Yu, and H. Lee, "Multiagent Q-learning-based multi-UAV wireless networks for maximizing energy efficiency: Deployment and power control strategy design," *IEEE Internet of Things Journal*, vol. 9, no. 9, pp. 6434-6442, May 2022. https://doi.org/10.1109/JIOT.2021.3113128

[14] D. Mandloi and R. Arya, "Q-learning-based UAV-mounted base station positioning in a disaster scenario for connectivity to the users located at unknown positions," *The Journal of Supercomputing*, vol. 79, pp. 15643–15674, 2023. https://doi.org/10.1007/s11227-023-05292-2

[15] S. Baghdady, S. M. Mirrezaei, and R. Mirzavand, "Reinforcement learning placement algorithm for optimization of UAV network in wireless communication," *IEEE Access*, vol. 12, pp. 37919-37936, 2024. https://doi.org/10.1109/ACCESS.2024.3374384

[16] A. Shamsoshoara, M. Khaledi, F. Afghah, A. Razi, J. Ashdown, and K. Turck, "A solution for dynamic spectrum management in mission-critical UAV networks," in *Proc. 16th IEEE Annual International Conference on Sensing, Communication, and Networking (SECON)*, 2019, pp. 1–6. https://doi.org/10.1109/SECON.2019.8824917

[17] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, Feb. 2015.

[18] X. Liu, Y. Chen, and L. Huang, "UAV trajectory optimization for data collection: A deep reinforcement learning approach," *IEEE Wireless Communications Letters*, vol. 9, no. 5, pp. 715–719, 2020. https://doi.org/10.1109/LWC.2020.2967123

[19] A. B. Bhandarkar, S. K. Jayaweera, and S. A. Lane, "User coverage maximization for a UAV-mounted base station using reinforcement learning and greedy methods," *in Proc. International Conference on Artificial Intelligence in Information and Communication (ICAIIC)*, pp. 351-356, 2022. https://doi.org/10.1109/ICAIIC54071.2022.9722679

[20] S. Yin and F. R. Yu, "Resource allocation and trajectory design in UAV-aided cellular networks based on multiagent reinforcement learning," *IEEE Internet of Things Journal*, vol. 9, no. 4, pp. 2933-2943, Feb. 2022. https://doi.org/10.1109/JIOT.2021.3094651

[21] B. Yin, X. Fang, and X. Wang, "Joint optimization of trajectory control, resource allocation, and user association based on DRL for multi-fixed-wing UAV networks," *IEEE Transactions on Wireless Communications*, vol. 23, no. 10, pp. 13330-13343, Oct. 2024. https://doi.org/10.1109/TWC.2024.3400821

[22] U. Challita, W. Saad, and C. Bettstetter, "Deep reinforcement learning for interference-aware path planning of cellular-connected UAVs," in *Proc. IEEE International Conference on Communications (ICC)*, 2018. https://doi.org/10.1109/ICC.2018.8422483

[23] H. Van Hasselt, A. Guez, and D. Silver, "Deep reinforcement learning with double Q-learning," *in Proc. AAAI Conf. Artif. Intell.*, vol. 30, no. 1, pp. 2094–2100, 2016. https://doi.org/10.1609/aaai.v30i1.10295

[24] R. Tang and Y. Zhou, "Utility maximization for multi-UAV-assisted IoT sensor systems with NOMA," *IEEE Wireless Communications Letters*, vol. 13, no. 11, pp. 3121–3125, Nov. 2024. https://doi.org/10.1109/LWC.2024.3452411