3D Trajectory Design of UAV Based on Deep Reinforcement Learning in Time-varying Scenes

Qingya Li
Liqy5523@bupt.edu.cn
School of Artificial Intelligence, Beijing University of Posts and Telecommunications, Beijing 100876
China
Key Laboratory of Universal Wireless Communications,Ministry of Education, Beijing University of Posts and Telecommunications,Beijing 100876
China

Li Guo*
guoli@bupt.edu.cn
School of Artificial Intelligence, Beijing University of Posts and Telecommunications, Beijing 100876
China
Key Laboratory of Universal Wireless Communications,Ministry of Education, Beijing University of Posts and Telecommunications,Beijing 100876
China

Chao Dong
dongchao@bupt.edu.cn
School of Artificial Intelligence, Beijing University of Posts and Telecommunications, Beijing 100876
China
Key Laboratory of Universal Wireless Communications,Ministry of Education, Beijing University of Posts and Telecommunications,Beijing 100876
China

Xidong Mu
muxidong@bupt.edu.cn
School of Artificial Intelligence, Beijing University of Posts and Telecommunications, Beijing 100876
China
Key Laboratory of Universal Wireless Communications,Ministry of Education, Beijing University of Posts and Telecommunications,Beijing 100876
China

ABSTRACT

A joint framework is proposed for the 3D trajectory design of an unmanned aerial vehicle (UAV) as an flying base station under the time-varying scenarios of users' mobility and communication request probability changes. The problem of 3D trajectory design is formulated for maximizing the throughput during a UAV's flying period while satisfying the rate requirement of all ground users (GUEs). Specifically, we consider that GUEs change their positions and communication request probabilities at each time slot; the UAV needs to predict these changes so that it can design its 3D trajectory in advance to achieve the optimization target. In an effort to solve this pertinent problem, an echo state network (ESN) based prediction algorithm is first proposed for predicting the positions and communication request probabilities of GUEs. Based on these predictions, a Deep Reinforcement Learning (DRL) method is then invoked for finding the optimal deployment locations of UAV in each time slots. The proposed method 1) uses ESN based predictions to represent a part of DRL agent's state; 2) designs the action and reward for DRL agent to learn the environment and its dynamics; 3) makes optimal strategy under the guidance of a double deep Q network (DDQN). The simulation results show that the UAV can dynamically adjust its trajectory to adapt to time-varying scenarios through our proposed algorithm and throughput gains of about 10.68% are achieved.

CCS CONCEPTS

• **Computing methodologies** → **Q-learning**.

KEYWORDS

UAV, Time-varying Scenes, Deep Reinforcement Learning (DRL), echo state network (ESN)

ACM Reference Format:
Qingya Li, Li Guo, Chao Dong, and Xidong Mu. 2021. 3D Trajectory Design of UAV Based on Deep Reinforcement Learning in Time-varying Scenes. In *2021 the 7th International Conference on Communication and Information Processing (ICCIP) (ICCIP 2021), December 16–18, 2021, Beijing, China.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3507971.3507982

1 INTRODUCTION

In recent years, given the high maneuverability, high mobility and low cost, unmanned aerial vehicles (UAVs) have shown great potential in the application of wireless communication systems[17]. Due to the high flying altitude, UAVs have a higher probability to establish Line-of-Sight (LoS) links compared to terrestrial communication links[1]. UAVs equipped with communication devices can be deployed as aerial base stations (BSs), which can enhance the coverage and performance of communication networks compared with the ground base stations (GBs).

Motivated by the above benefits of UAVs, growing research efforts have been devoted, which can be loosely divided into two categories, namely deployment design and trajectory design. On the one hand, some papers have investigated the optimal deployment

*Li Guo is corresponding author.

ICCIP 2021, December 16–18, 2021, Beijing, China

ACM ISBN 978-1-4503-8519-0/21/12…$15.00
https://doi.org/10.1145/3507971.3507982

location of UAV BSs [2, 8, 10, 11]. The authors of[10] optimized the horizontal positions of UAVs to minimize the number of required UAV BSs to cover a given set of ground users (GUEs) by fixing the altitude. An optimum placement of multiple UAVs in three dimensional (3D) space is studied in [2] to maximize the number of covered users in the target region. In[11], UAV-aided devide to device (D2D) communications was investigated, the UAV altitude is appropriately adjusted based on the D2D users density and a trade-off between the coverage area and the time required for covering the entire target area (delay) by UAV-aided data acquisition was also analyzed. On the other hand, the problem of UAV trajectory design and optimization is studied[5, 12, 14–16]. The authors of [14] investigated the trajectory design in a Multi-UAV BSs network. In order to maximize the minimum throughput over all ground users in the downlink communication, the UAV's trajectory, user scheduling and UAVs transmit power are optimized. The authors of [16] apply sequential convex optimization techniques to solve the non-convex trajectory optimization problems that jointly considers both the communication throughput and the UAV's energy consumption.

Reinforcement learning (RL) can learn good policies for sequential decision problems by optimizing a cumulative future reward signal, which becomes a promising technique of improving the control of UAV networks. The authors of [8] proposed a Q-learning based algorithm for obtaining 3D dynamic movement of UAVs. Deep reinforcement learning (DRL) has been proved to be more suitable for solving non-convex problems and high-dimension solutions compared with RL,relying on the powerful function approximation and representation learning properties of deep neural network (DNN) [12]. The authors in [5] invoked a DRL-based method to find a control strategy that specifies how each UAV moves in each time slot to meet the limitations of communication coverage, fairness, energy consumption and connectivity.

Time-varying factors of GUEs have been introduced into UAV aided communication scenario recently. Considering the mobility of GUEs, the authors in [6] design the placement and movement of the UAVs to achieve maximum average throughput with full moving users' real location information from record history data and authors of [7] simply assumes that the user moves randomly. The authors of[9] proposed a pattern formation based framework in a machine learning manner to track the instable and non-ergodic time-varying nature of user density.

As mentioned above, although the control of UAV BS has received extensive attention from academia and industry, few of current research contributions consider adopting DRL method to design the UAV's 3D trajectory under the time-varying scenarios of GUEs' mobility and communication request probability changes. Moreover, there is still a paucity of research contributions on predicting the mobility and the communication request probability by time series under the UAV aided communication networks, which motivates this treatise.

In this paper, the trajectory of the UAV BS to provide uplink communication service for GUEs is optimized, taking into account the minimum rate requirement of GUEs, user's mobility and communication request probability. As shown in Fig.1, we proposed an ESN-DDQN joint framework to solve this complex problem. The

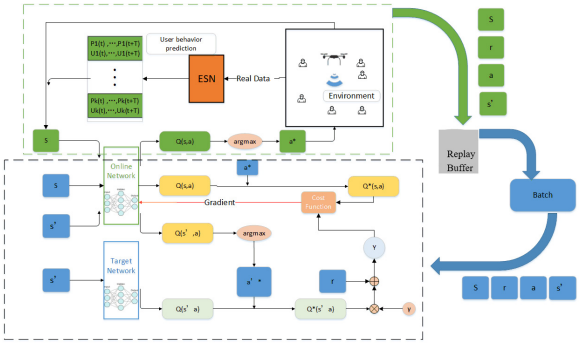


Figure 1: ESN-DDQN Joint Framework.

ESN-based algorithm predict the communication request probability and movement of each GUE, and these predictions are used as the input part of the DDQN to train the agent to design the trajectory of UAV BS. Experimental results show that the UAV BS can adapt to the time-varying characteristics of the GUEs, and achieves a long-term gain in throughput.

2 SYSTEM AND PROBLEM FORMULATION

2.1 System Model

We consider a UAV-assisted wireless network, where a UAV BS dynamically adjusts its 3D trajectory to provide uplink communication services for a group of K GUEs to collect information in a target area. During the working time period T of the UAV BS, we consider UAV communicates simultaneously with all the users, while every GUE continuously moves and changes the probability of communication requests. The flying time of the UAV BS is discretized into N_T equal-length time slots, and each time slot is small enough that we believe the position and communication request probability of the GUEs are fixed during one time slot. Ignore the height of GUEs, the coordinate of each GUE can be expressed as $u_k(t) = [x_k(t), y_k(t)]; k \in K$, where $x_k(t)$ and $y_k(t)$ are the X-coordinate and Y-coordinate of user k at time slot t, respectively. The movement of the GUE is recorded by a polyline generated by $u_k(t)$ and the probability communication request is expressed as

$$p_k(t) = \tau_k(t)/\delta T, \tag{1}$$

where $\tau_k(t)$ represents the accumulated communication time between user k and UAV BS during the t-th time slot, and $\delta T = T/N_T$ represents the duration of one time slot. The UAV needs to adjust its deployment location at each flight slot to adapt to the changes of GUEs, and these locations form a 3D trajectory denoted by $\mathbf{q(t)=[x(t),y(t),h(t)]}; 0 < t < N_T$. The coordinate of the UAV is obtained through the GPS device carried by itself, and since the UAV has a high flight speed, the UAV takes negligible time to adjust its location between the two time slots.

In our model, the channel condition between UAV and GUEs can be regarded as air-to-ground channel, in which the LoS condition and non-line-of-sight (NLoS) condition appear randomly. The LoS probability can be expressed as [2]

$$P_k^{\mathrm{LoS}} = \frac{1}{1 + aexp(-b(\frac{180}{\pi} tan^{-1}(\theta_k(t)) - a))}, \tag{2}$$

where $\theta_k(t)$ is the angle between the UAV and the user k at t-th time slot, a and b are parameters related with the environment. Naturally, the (NLoS) probability can be derived as $P_k^{\mathrm{NLoS}} = 1 - P_k^{\mathrm{LoS}}$. The distance between the UAV and the GUE k during time slot t can be expressed as

$$d_k(t) = \sqrt{h(t)^2 + (x(t) - x_k(t))^2 + (y(t) - y_k(t))^2}, \tag{3}$$

Therefore, the channel power gain between the UAV and user k at time slot t is given by

$$g_k(t) = K_0^{-1} d_k^{-\alpha}(t)[P_k^{\mathrm{LoS}} \mu_{LoS} + P_k^{\mathrm{NLoS}} \mu_{NLos}]^{-1}, \tag{4}$$

where $K_0 = (\frac{4\pi f_c}{c})^2$, α is the path loss exponent, μ_{LoS} and μ_{NLos} are the mean extra losses for the LoS and NLoS respectively, f_c represents the carrier frequency, and c is the speed of light.

We assume frequency division multiple access (FDMA) to serve users, the total available bandwidth B of the UAV is equally distributed to all GUEs, so the bandwidth of each user can be expressed as $B_k = B/K$. And rate of user k at time slot t can be expressed as

$$r_k(t) = p_k(t) \cdot B_k log(1 + \frac{P \cdot g_k(t)}{\sigma_2}), \tag{5}$$

where $p_k(t)$ denotes the communication request probability of user k at time slot t, P denotes uplink transmit power of GUE, $\sigma^2 = B_k N_0$ with N_0 denoting the power spectral density of the additive white Gaussian noise (AWGN) at the UAV BS.

2.2 Problem Formulation

We consider the Quality of service (QoS) of the GUEs by guaranteeing the minimum transmission rate r_0 for each user. Let $Q = q(t); 0 \le t \le N_T$. Our optimization problem is then formulated as

$$\max_Q R_{total}^T = \sum_{t=0}^{T} \sum_{k=1}^{K} r_k(t) \tag{6}$$

$$\textbf{s.t. } H_{min} \le h(t) \le H_{max}, 0 \le t \le N_T, \tag{7}$$

$$X_{min} \le x(t) \le X_{max}, 0 \le t \le N_T, \tag{8}$$

$$Y_{min} \le y(t) \le Y_{max}, 0 \le t \le N_T, \tag{9}$$

$$r_k(t) \ge r_0, \forall k, t. \tag{10}$$

We aim to maximize the total throughput R_{total}^T over a long UAV flight period, during which the communication request probability and position of GUEs may change multiple times. Formula (7), (8) and (9) represent the flight space limit of the UAV. Formula (10) is introduced to limit the minimum communication rates for every user. The optimization target R_{total}^T is determined not only by the UAV 3D trajectory but the time-varying factors of the GUEs. However, these time-varying factors of GUEs are not optimizable, they change objectively over time. In the process of optimizing UAV trajectory, the influence of these time-varying factors must be taken into account, which makes the problem more difficult to solve. To address these challenges, we first used ESN to predict time-varying factors. With these time-varying factors known, the problem (6) is still challenging since the objective function is non-convex as a function of $x(t)$, $y(t)$ and $h(t)$. Since we can't obtain a feasible solution by general methods to solve this non-convex problem, then the DRL based strategy is proposed.

3 ESN-BASED PREDICTION ALGORITHM

The optimization target in (6) is affected by the GUEs' communication request probabilities and locations, so UAV must get these information in advance so it can plan its 3D trajectory. To solve this problem, we proposed an algorithm based on the powerful framework of echo state networks to predict these user behaviors.

In our model, we use vectors to represent the locations and communication request probabilities of GUEs, these vectors can be taken as time series, which can be well predicted by ESN[3]. The ESN model sequentially consists of three layers:

- Input: $u_{k,t}$ and $p_{k,t}$ respectively represent the position and communication request probability of user k during the t-th time slot.
- Neuron reservoir: The neuron reservoir is a network composed of N_x sparse neurons with invariable connection weight and haves a short term memory of the previous states encountered.
- Output: $u_{k,t+1}$ and $p_{k,t+1}$ represent the position and communication request probability of user k that is predicted in the next time slot.

The dynamic reservoir state update equation of ESN with N_x reservoir units, L_{in}-dimensional inputs and L_{out}-dimensional outputs is governed by

$$x(n+1) = f_1(W \cdot x(n) + W_{in} \cdot u(n+1)), \tag{11}$$

where f_1 is a sigmoid function, $x(n)$ denotes N_x-dimensional reservoir state, W denotes the $N_x \times N_x$ reservoir weight matrix, W_{in} denotes the $N_x \times L_{in}$ input weight matrix, $u(n)$ is the L_{in}-dimensional input vector. The output is obtained from the extended system state $z(n) = [x(n) : u(n)]$ by

$$y(n) = W_{out} \cdot z(n), \tag{12}$$

where $y(n)$ denotes L_{out}-dimensional output vector, and W_{out} is a $L_{out} \times (L_{in} + N_x)$-dimensional matrix of output weights.

The matrix W_{in}, W and W_{out} are randomly initialized, and only the W_{out} needs to be trained by

$$\min \frac{1}{N} \sum_{n=N_I}^{N_I+N} ||Y_{target}(n) - y(n)||^2, \tag{13}$$

where $Y_{target}(n)$ represents the real data, N represents the length of training data. In our model, we initialize reservoir state firstly with N_I data to reduce the influence of noise caused by the the random connections within Neuron reservoir. The size of neuron reservoir N_x is a key parameter to determine the memory capacity of ESN model, which directly affects the accuracy of the prediction; we discuss this parameter N_x in section V. The specific implementation process is explained in Algorithm 1.

Algorithm 1 ESN based prediction algorithm for communication request probability and movement of GUEs

Initialize: Length of predicting data L, length of training data N_p for ESN_1 and N_u for ESN_2, length of initial data N_I.
for user $k = 1, \cdots, K$ **do**
  Calculate the communication request probability vector $[p_{k,1}, \cdots, p_{k,N_p}]$ according to formula (1).
  Initialize the parameter W, W_{in}, W_{out} of ESN_1.
  for $i = 1, \cdots, N_p + N_I$ **do**
    Update reservoir state according to Eq.(11) and outputs according to Eq.(12).
  end for
  Calculate W_{out} of ESN_1 according to formula(13).
  Get the prediction of users'communication request probability vector $[p_{k,N_p+1}, \cdots, p_{k,N_p+L}]$.
  Calculate the location vector $[u_{k,1}, \cdots, u_{k,N_u}]$ based on real data.
  Initialize the parameter W,W_{in},W_{out} of ESN_2.
  for $i = 1, \cdots, N_u + N_I$ **do**
    Update reservoir state according to Eq.(11) and outputs according to Eq.(12).
  end for
  Calculate W_{out} of ESN_2 according to formula(13).
  Get the prediction of users' movement vector $[u_{k,N_u+1}, \cdots, u_{k,N_u+L}]$.
end for

4 DRL FOR 3D TRAJECTORY OPTIMIZATION OF UAV IN TIME-VARYING SCENES

4.1 DDQN Method

Before introducing the proposed method, we first provide some background on DRL in this subsection.

A RL problem can be described as a MarKov Decision Process (MDP); in its setting, an agent interacts with a system environment and makes a series of decisions in discrete epochs. At each epoch t, the agent observes state s_t, executes action a_t, receives a reward r_t, and transits to the next state s_{t+1}. The purpose of RL is to find an optimal policy $\pi(s, a)$, which maps a state to an action for maximizing the discounted cumulative expectation of rewards

$$Q^{\pi}(s, a) = \sum_{k=0}^{T} \gamma^k r_{t+k} | (s_t, a_t, \pi), \tag{14}$$

where $\gamma \in [0, 1]$ is the discount factor.

In this paper, we adopt an algorithm based on DDQN, and it has been proved to be able to overcome over estimations of action values by decomposing the max-Q-value operation into action selection and action evaluation compared with deep Q-network (DQN) algorithm[13]. As shown in Fig.1, DDQN framework has two deep Q-networks with the same architecture, defined as the Q online network and the Q target network. The online network is used for generating the $Q(s, a)$ with weights θ. For each step of agent, $Q(s, a)$ is used to choose action a by the greedy policy:arg max $Q(s, a)$. However, this action's Q value denotes $Q'(s, a)$ is determined by the target network with weights θ'. The online network is trained to optimize parameters θ by means of optimizing the loss function.

$$L_{oss}(\theta) = E[(y - Q(s, a; \theta)^2], \tag{15}$$

where y is the target Q value which can be denoted by

$$y = r + \gamma \max_{a'} Q'(s', a'; \theta'), \tag{16}$$

as for the target network, its parameters θ' are copied every w steps from online network, so that then $\theta = \theta'$, and keep fixed on all other steps.

Experience replay is a useful strategy of DDQN. The agent store the tuple $[s, a, r, s']$ into replay buffer, and randomly sample M_b samples for training the deep Q networks. With experience replay, essential historical information can be stored and used again to train deep Q networks, which can speed up the convergence of the algorithm.

4.2 DDQN-Based 3D Trajectory Design Algorithm

In our model, ESNs are invoked to predict each GUE's position and communication request probability in advance at each update time slot, and then these predictions can be thought of as invariant during this time slot. With each time slot update, we need to retrain the agent to adapt to these new predictions of GUEs. To find the optimal deployment location in this update time slot, the agent needs to be iteratively trained E times, and we define I represents the maximum step length per episode. During each step, the UAV executes the action a according to the current strategy π, and the current state s turns to the next state s'. Then, let's define the state, action and reward function of the agent.

1) State $s_t(i)$ (during t-th time slot, in the i-th training step). $s_t(i)$ consists of three parts:

- $[u_{1t}, \cdots, u_{kt}]$:The position of the GUEs at t-th time slot. u_{kt} is a two-dimensional vector that represents the X and Y coordinates of user k.
- $[p_{1t}, \cdots, p_{kt}]$:The communication request probability of the GUEs at t-th time slot.
- $[x_t(i), y_t(i), h_t(i)]$:During t-th time slot, the location of the UAV in the i-th training step.

Then the state $s_t(i) = [u_{1t}, \cdots, u_{kt}, p_{1t}, \cdots, p_{kt}, x_t(i), y_t(i), h_t(i)]$, which has a cardinality of $(3K + 3)$. Note that the state is defined in this way because the DRL agent makes decisions mainly based on the GUEs' position and communication request probability and the UAV's location.

2) Action $a_t(i)$ (during t-th time slot, in the i-th training step). We consider that the UAV has seven flight directions and we use a three-dimensional vector to represent.Explicitly, (1, 0, 0) means right; (−1, 0, 0) indicates left; (0, 1, 0) represents forward; (0, −1, 0) means backward; (0, 0, 1) implies rises; (0, 0, −1) means descends; (0, 0, 0) indicates static.

3) Reward $R_t(i)$ (during t-th time slot, in the i-th training step): the reward $R_t(i)$ is defined as

$$R_t(i) = \begin{cases} Sumrate - LastSumrate, & \forall r_k(t) \geq r_0, \\ -LastSumRate, & \exists r_k(t) < r_0, \end{cases} \tag{17}$$

where Sumrate represents the throughput of the UAV BS in the current time slot and LastSumrate denotes throughput of last time

slot. We use the difference between the throughput of two adjacent time slots to indicate agent's progress or regression. When the minimum requirement transmission rate of each GUE cannot be satisfied, the throughput of the UAV is set to zero; then the agent receives a large penalty.

In our DDQN model, we used a 2-layer fully connected neural network with 64 and 32 neurons respectively, and utilized the ReLU function for activation. The specific operation steps is formally presented as Algorithm 2.

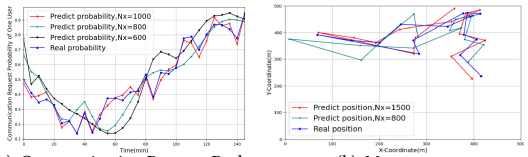
Algorithm 2 DDQN-based 3D trajectory design algorithm

Initialize: The replay buffer size M, the online network $Q(s,a;\theta)$ with weights θ and target network $Q'(s,a;\theta')$ with parameters θ'.
for Time slots $n = 1, \cdots, N$ **do**
  for Episode=$1, \cdots, E$ **do**
    for Step $i = 1, \cdots, I$ **do**
      Observe current state $s_n(i)$.
      Choose action with ϵ-greedy policy: $a_n(i) = \arg\max(Q_n(s_t(i), a))$, where ϵ increase over episode.
      Receive reward $R_n(i)$.
      Observe next state $s_n(i+1)$.
      Store $[s_n(i), a_n(i), R_n(i), s_n(i+1)]$ into the replay buffer.
      Randomly select a minibatch of M_b samples from replay buffer.
      Train the online network, and update the weights θ.
      In every w steps,let $\theta' = \theta$.
    end for
  end for
  UAV flies directly to the optimal position for time slot t, forming a 3D trajectory.
end for

5 SIMULATION RESULTS

In this section, we present numerical results to verify the effectiveness and superiority of the proposed algorithm in solving 3D trajectory design problems of UAV under the time-varying scenarios of ground user request probability and location.

In our simulation, we set the duration of each time slot as 5 minutes, and have simulated $T = 8$ time slots. The flight boundary of the UAV is $500m \times 500m$, and the altitude ranges 120 m to 180 m. We set the number of GUEs as $K = 10$, and other simulation parameters are presented in Table I.



(a) Communication Request Probability (b) Movement

Figure 2: User behaviors Prediction

Table 1: Simulation parameters

Parameter	Description	Value
F_c	Carrier frequency	2Ghz
P	Transmission power	0.2W
N_0	Noise power spectral	−120dBm/hz
B	Bandwidth	1Mhz
a,b	Environmental parameters	9.61,0.16
α	Path loss exponent	2
μ_{Los}	Additional path loss for Los	1dB
μ_{NLos}	Additional path loss for NLos	20dB
γ	Discount factor	0.9
E	Episodes	300
I	Step length	100
M_b	Mini-batch size	32
M	Replay buffer size	10000
w	Update frequency	600
r_0	Minimum rate requirement	0.4bits/s/hz

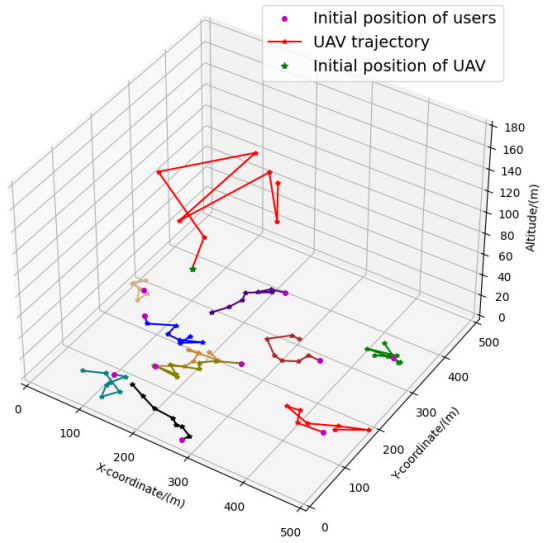


Figure 3: The designed trajectory of UAV and predicted trajectories of GUEs

Fig.2 characterizes one user's behaviors from the validation set and the predicted results by ESNs. The data used by the ESN algorithm is based on the time series obtained by simulation. ESNs are used to predict the communication request probability and location movement of GUEs separately, and we set N_I to 100, N_p to 1500, and N_u to 2000. we can see that the proposed ESN approach achieves more improvement in terms of the prediction accuracy with the increase of N_x.

Fig.3 shows the UAV trajectory and the predicted trajectories of 10 ground users in 8 time slots. The UAV designed trajectory is formed according to the optimal deployment locations of these 8 time slots. In each time slot, DDQN based algorithm is used to train

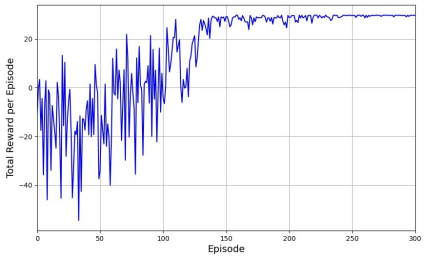


Figure 4: Total reward versus training episodes.

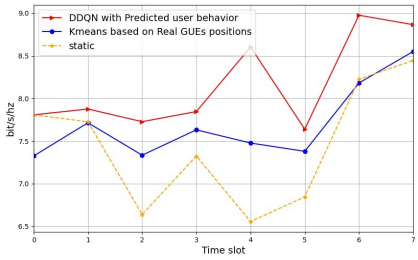


Figure 5: Comparison between contrast schemes and proposed method over throughput

the agent to find these optimal deployment locations to adapt to the time-varying factors of GUEs.

Fig.4 characterizes the training process of the agent by DDQN based algorithm in one time slot. This method is evaluated with learning rate $L_r = 0.001$ and AdamOptimizer optimization strategy. With the increase of training episodes, the reward of the agent gradually increases and finally converges. The process of the reward curve grows and eventually converges with the training episodes implies the UAV is trained to find the optimal deployment location.

Two contrast schemes were introduced to verify the performance of our methods from various perspectives. In first scheme,K-means algorithm is capable of obtaining the deployment location based on the policy of nearest neighbor barycenter[4], and this scheme doesn't consider the user's request probability. Scheme 2 named static deployment, adopts DDQN based algorithm to optimize deployment location with the initial user behaviors in first time slot, while it can't predict changes of user behaviors and keeps UAV's location static in next time slots.

Table 2: Number of users guaranteed the minimum communication rate

Scheme \ Time slot	0	1	2	3	4	5	6	7
ESN-DDQN	10	10	10	10	10	10	10	10
Kmeans	10	10	10	10	10	9	10	9
Static	10	10	10	9	10	9	8	9

Fig.5 characterizes the throughput generated by our proposed ESN-DDQN joint framework algorithm and two contrast schemes in 8 time slots. Overall,the proposed algorithm achieves an average 7.10% improvement compare with K-means algorithm and 10.68% improvement compare with static scheme in total throughput. Without considering the factor of the communication request probability changes, throughput generated by Kmeans algorithm is still lower than our proposed method even the real location information of GUEs form the validation set can be used. As for static scheme, it adopts DDQN based algorithm with initial user's location and communication request probability information to deploy UAV, so the throughput in first time slot is better than K-means algorithm. However, UAV BS keeps static next time slots, then it generates the worst throughput performance in next time slots. As shown in Table II, our scheme guarantees a minimum communication rate for all users, while the other two schemes do not perform well.

6 CONCLUSIONS

The 3D trajectory of the UAV BS was designed based on the proposed joint ESN-DDQN framework to adapt to the time-varying scenarios of communication request probability and mobility of GUEs. Firstly, we adopted ESN to predict the user's position and communication request probability in each time slot, and then took these prediction as part of the input of DDQN. DDQN trained the agent to find the optimal deployment position of the UAV in each time slot, and finally formed the trajectory in the whole flight time period. Through our proposed method, the UAV can dynamically adjust its trajectory according to the predicted user behavior. Numerical results reveal that the proposed 3D trajectory design scheme achieved a 10.68% gains compared with static scheme and a 7.10% gains compared with K-means algorithm in throughput.

ACKNOWLEDGMENTS

This work is supported by the Beijing Natural Science Foundation (No. L192032), National Key Research and Development Program of China (No. 2019YFB1406500), and the Key Project Plan of Blockchain in Ministry of Education of the People's Republic of China under Grant No.2020KJ010802.

REFERENCES

[1] A. Al-Hourani, S. Kandeepan, and A. Jamalipour. 2014. Modeling air-to-ground path loss for low altitude platforms in urban environments. In *2014 IEEE Global Communications Conference*. 2898–2904. https://doi.org/10.1109/GLOCOM.2014.7037248

[2] R. I. Bor-Yaliniz, A. El-Keyi, and H. Yanikomeroglu. 2016. Efficient 3-D placement of an aerial base station in next generation cellular networks. In *2016 IEEE International Conference on Communications (ICC)*. 1–5. https://doi.org/10.1109/ICC.2016.7510820

[3] M. Chen, W. Saad, C. Yin, and M. Debbah. 2017. Echo State Networks for Proactive Caching in Cloud-Based Radio Access Networks With Mobile Users. *IEEE Transactions on Wireless Communications* 16, 6 (2017), 3520–3535. https://doi.org/10.1109/TWC.2017.2683482

[4] T. Kanungo, D. M. Mount, N. S. Netanyahu, C. D. Piatko, R. Silverman, and A. Y. Wu. 2002. An efficient k-means clustering algorithm: analysis and implementation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24, 7 (2002), 881–892. https://doi.org/10.1109/TPAMI.2002.1017616

[5] C. H. Liu, Z. Chen, J. Tang, J. Xu, and C. Piao. 2018. Energy-Efficient UAV Control for Effective and Fair Communication Coverage: A Deep Reinforcement Learning Approach. *IEEE Journal on Selected Areas in Communications* 36, 9 (2018), 2059–2070. https://doi.org/10.1109/JSAC.2018.2864373

[6] L. Liu, S. Zhang, and R. Zhang. 2019. CoMP in the Sky: UAV Placement and Movement Optimization for Multi-User Communications. *IEEE Transactions on Communications* 67, 8 (2019), 5645–5658. https://doi.org/10.1109/TCOMM.2019.2907944

[7] X. Liu, Y. Liu, and Y. Chen. 2018. Deployment and Movement for Multiple Aerial Base Stations by Reinforcement Learning. In *2018 IEEE Globecom Workshops (GC Wkshps)*. 1–6. https://doi.org/10.1109/GLOCOMW.2018.8644345

[8] X. Liu, Y. Liu, and Y. Chen. 2019. Reinforcement Learning in Multiple-UAV Networks: Deployment and Movement Design. *IEEE Transactions on Vehicular Technology* 68, 8 (2019), 8036–8049. https://doi.org/10.1109/TVT.2019.2922849

[9] J. Lu, S. Wan, X. Chen, Z. Chen, P. Fan, and K. B. Letaief. 2018. Beyond Empirical Models: Pattern Formation Driven Placement of UAV Base Stations. *IEEE Transactions on Wireless Communications* 17, 6 (2018), 3641–3655. https://doi.org/10.1109/TWC.2018.2812167

[10] J. Lyu, Y. Zeng, R. Zhang, and T. J. Lim. 2017. Placement Optimization of UAV-Mounted Mobile Base Stations. *IEEE Communications Letters* 21, 3 (2017), 604–607. https://doi.org/10.1109/LCOMM.2016.2633248

[11] M. Mozaffari, W. Saad, M. Bennis, and M. Debbah. 2016. Unmanned Aerial Vehicle With Underlaid Device-to-Device Communications: Performance and Tradeoffs. *IEEE Transactions on Wireless Communications* 15, 6 (2016), 3949–3963. https://doi.org/10.1109/TWC.2016.2531652

[12] J. Tang, J. Song, J. Ou, J. Luo, X. Zhang, and K. Wong. 2020. Minimum Throughput Maximization for Multi-UAV Enabled WPCN: A Deep Reinforcement Learning Method. *IEEE Access* 8 (2020), 9124–9132. https://doi.org/10.1109/ACCESS.2020.2964042

[13] Hado van Hasselt, Arthur Guez, and David Silver. 2015. Deep Reinforcement Learning with Double Q-learning. arXiv:1509.06461 [cs.LG]

[14] Q. Wu, Y. Zeng, and R. Zhang. 2018. Joint Trajectory and Communication Design for Multi-UAV Enabled Wireless Networks. *IEEE Transactions on Wireless Communications* 17, 3 (2018), 2109–2121. https://doi.org/10.1109/TWC.2017.2789293

[15] D. Yang, Q. Wu, Y. Zeng, and R. Zhang. 2018. Energy Tradeoff in Ground-to-UAV Communication via Trajectory Design. *IEEE Transactions on Vehicular Technology* 67, 7 (2018), 6721–6726. https://doi.org/10.1109/TVT.2018.2816244

[16] Y. Zeng and R. Zhang. 2017. Energy-Efficient UAV Communication With Trajectory Optimization. *IEEE Transactions on Wireless Communications* 16, 6 (2017), 3747–3760. https://doi.org/10.1109/TWC.2017.2688328

[17] Y. Zeng, R. Zhang, and T. J. Lim. 2016. Wireless communications with unmanned aerial vehicles: opportunities and challenges. *IEEE Communications Magazine* 54, 5 (2016), 36–42. https://doi.org/10.1109/MCOM.2016.7470933