Learning-based Localization of Mobile Users for Throughput Maximization in UAV Networks

Arzhang Shahbazi, Marco Di Renzo
Laboratoire des Signaux et Systèmes, University of Paris-Saclay, CNRS, CentraleSupélec, Gif-Sur-Yvette, France
arzhang.shahbazi@centralesupelec.fr, marco.direnzo@centralesupelec.fr

***Abstract*—In this paper, we design a new UAV-assisted communication system relying on the shortest flight path of the UAV while maximising the amount of data transmitted to mobile devices. In the considered system, we assume that UAV does not have the knowledge of user's location except their initial position. We propose a framework which is based on the likelihood of mobile users presence in a grid with respect to their probability distribution. Then, a deep reinforcement learning technique is developed for finding the trajectory to maximize the throughput in a specific coverage area. Numerical results are presented to highlight how our technique strike a balance between the throughput achieved, trajectory, and the complexity.**

***Index Terms*—Mobility, throughput, reinforcement learning, unmanned aerial vehicles**

I. Introduction

Unmanned aerial vehicles (UAVs) have recently captivated interest as a rapid solution for providing communication services to ground users [1], [2]. In practice, it is not cost-effective or even practical to set up terrestrial base stations (BSs) in temporary hotspots or disaster areas. In contrast, due to the exceptional flexibility of deployment and maneuverability of UAVs, they can be employed in an efficient manner to serve as aerial BSs [3]. Moreover, the communication link between users and UAVs has typically high probabilities of line-of-sight (LoS) air-to-ground (A2G) channels, which can mitigate signal blockage and shadowing [4] . Wireless networks supported by UAVs constitute a promising technology for enhancing the network performance [5]. The applications of UAVs in wireless networks span across diverse research fields, such as wireless sensor networks (WSNs), caching, heterogeneous cellular networks, massive multiple-input multiple-output (MIMO), disaster communications and device-to-device communications (D2D). In all mentioned scenarios, a critical aspect for the system's ability to serve the highest possible number of users with the best achievable throughput is the user's location. Previous works have addressed the problem of path planning of UAV by neglecting the mobility of users in to the system model. Whereas fixed location of users may fulfill certain communication network scenarios, but in real life applications, one can not oversight the dynamic movement of users. In [6], the authors studied the joint 3D deployment and power allocation in a UAV-BS system that maximizes the system throughput. They proposed an algorithm which combined deep deterministic policy gradient with water-filling to allow the UAV to learn an optimal location in the continuous state and action spaces. In [7], the authors investigated the multi-UAV trajectory planning to provide a long-term energy-efficient content coverage. A multi-UAV trajectory planning problem was formulated as two related multi-agent cooperative stochastic games. For obtaining equilibriums of the games, the authors proposed a Q-learning based decentralized multi-UAV cooperative RL algorithm. The proposed algorithm enables UAVs to independently choose their policy and recharging scheduling. Also, in a decentralized manner, the UAVs share their learning results with each other over a timevarying communication network. In [8], authors proposed a 3D deployment based on the quality of experience and they considered the dynamic movement of ground users into their system model. They demonstrated that the proposed 3D deployment scheme based on Q-learning outperforms the K-means algorithm. However, the authors assumed the UAV has online knowledge of dynamic movement of ground users which is not always possible in real life applications.

In this paper, we consider a system model relying on a single UAV to serve several mobile users. We propose a framework for finding the trajectory to maximize the achievable system throughput between all users. In our proposed model, the UAV is only aware of the initial position of users and needs to choose actions based on the stochastic model calculated from the mobility of users. For comparison, we consider a scenario that UAV is connected through the GPS system and has the knowledge of user's location in each time instant.

The paper is organized as follows: the system model and achievable system throughput are given in section II. In Section III, mobility model and stochastic model for localization of users are proposed. In Section IV, the deep reinforcement learning algorithm is utilized for obtaining the UAVs' dynamic movement when users are roaming. Numerical results are carried out in Section V. Finally, the paper is concluded in Section VI.

II. System Model

Consider a system consisting of a single UAV and U ground users with dynamic movement in the area and need to be covered. Let $\boldsymbol{u}_u = [x_u, y_u]^T \in \mathbb{R}^{2\times 1}$ represent the horizontal coordinate of u-th ground user where $u \in \boldsymbol{U}$. The 2D Cartesian coordinate of the UAV is presented as

This work was supported in part by the European Commission through the H2020 PAINLESS Project under Grant 812991 and the H2020 ARIADNE Project under Grant 871464.

$m = [x_m, y_m]^T$. In practice, the ground users receive three different kinds of signals from UAVs including LoS, non-line-of-sight (NLoS), and multiple reflected signals. These signals occur with specific probabilities in different environments and the probability of multiple reflected signal which results multi-path fading is considerably lower than two other signals. Thus, their impact at the receiver side is typically ignored. Thus, we assume that the communication link between ground users and the UAV is overshadowed by the LoS signals. Based on this assumption, the channel power gain between u-th user and the UAV is only a function of their Euclidean distance as below

$$h_{u,m} = \rho_0 d_{u,m}^{-2} \tag{1}$$

where ρ_0 is a constant shadowing power of the channel at the reference distance $d_0 = 1$m and d_u is the Euclidean distance between u-th user and UAV which can be written as

$$d_u = \sqrt{z_m^2 + \|u_u - m\|} \tag{2}$$

Hence, we have

$$h_{u,m}(t) = \frac{\rho_0}{z_m^2 + \|u_u - m\|} \tag{3}$$

The bit rate at time t for u-th user can be formulated as below

$$R_u(t) = \log_2(1 + \gamma_{u,m}(t)) \tag{4}$$

where $\gamma_u(t)$ is the signal-to-noise ratio (SNR) corresponding to the u-th user at time t, which can be expressed as

$$\gamma_{u,m}(t) = \frac{P h_{u,m}(t)}{\sigma^2} \tag{5}$$

where P is the UAV transmit power and σ^2 is the power of the additive white Gaussian noise (AWGN) at u-th user. Since users are mobile, for each user, there are k possible locations with respect to time. So we have

$$Pr_u^{(x_k,y_k)}(t) = z \quad \forall u, \forall t, \forall u \tag{6}$$

Consequently, by utilising the above probability, the achievable system throughput can be expressed as

$$R_k^{(x_k,y_k)}(t) = \sum_{x_k,y_k} Pr_u^{(x_k,y_k)}(t) \times R_u^k(t) \tag{7}$$

Since the movement of users affect the system throughput, the UAV have to travel based on the real-time movement of users to maximize the throughput for ground users. Thus, to provide communication services for all ground users, we maximize the achievable system throughput subject to the location of each user based on their mobility model. So, we can write

$$\max_{x_m(t),y_m(t)} \left(\int_{t=0}^{T} \sum_{u=1}^{U} R_u^k(t) dt \right) \tag{8}$$

$$\text{s.t.} \quad x_1(0), ..., x_u(0) = X_1(0), ..., X_u(0), \forall u \tag{9}$$

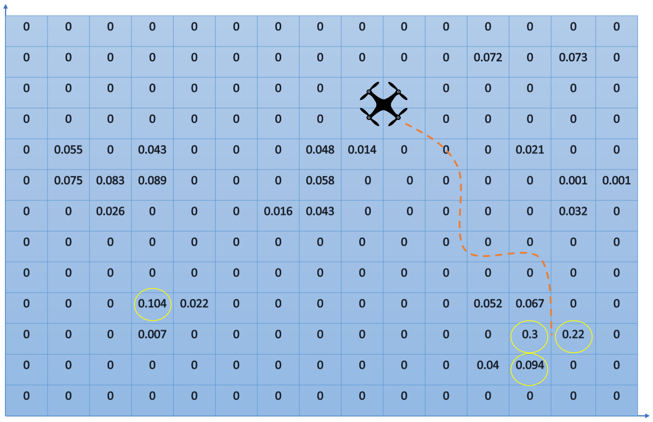
$$y_1(0), ..., y_u(0) = Y_1(0), ..., Y_u(0), \forall u \tag{10}$$

$$x_u^k(t), y_u^k(t) = Pr_u^{(x_k,y_k)}(t), \forall k, \forall t, \forall u \tag{11}$$

$$z_m(t) = H_{uav} \tag{12}$$

$$P_{tx}(t) = P_m \tag{13}$$

$$V_c(t) = V_{uav} \tag{14}$$



0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0.072	0	0.073	0
0	0	0	0	0	0	0	0			0	0	0	0	0
0	0	0	0	0	0	0	0			0	0	0	0	0
0	0.055	0	0.043	0	0	0	0.048	0.014	0	0	0	0.021	0	0
0	0.075	0.083	0.089	0	0	0	0.058	0	0	0	0	0	0.001	0.001
0	0	0.026	0	0	0	0.016	0.043	0	0	0	0	0	0.032	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0.104	0.022	0	0	0	0	0	0	0.052	0.067	0	0
0	0	0	0.007	0	0	0	0	0	0	0	0	0.3	0.22	0
0	0	0	0	0	0	0	0	0	0	0	0.04	0.094	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fig. 1: Probability distribution of a mobile user based on the grid model.

where H_{uav} and V_{uav} are the altitude and velocity of UAV, while P_c is the value for transmit power from UAV to ground users. Furthermore, (9) and (10) denote that initial position of each user is known by the UAV; (11) indicates that the location of mobile users are estimated based on their probability distribution, (12),(13) and (14) set the constant values on altitude, transmit power and velocity of the UAV, respectively.

III. MOBILITY MODEL AND PROBABILITY DISTRIBUTION OF USER'S LOCATION

The memoryless mobility models such as Random Walk allow mobile nodes to move anywhere in the system with a stochastic random process for speed and direction. Consequently, the mobility patterns are very disordered and may not be able to reflect the real-time scenarios of mobile ad hoc networks. In reality, movements of mobile nodes are restricted by obstacles. Moreover, there is some correlation between the speed, direction, path, and destination of mobile nodes to meet their corresponding objectives.

Since our objective is to let the UAV learn the trajectory based on the mobility of users, the choice of the mobility model has a major impact on the learned trajectory. If we consider a model that users change their direction or speed at each time step, the randomness in the environment is too chaotic in which, there is no meaningful trajectory to be learned. Also, border behavior of the environment and how users react when they reach the border cannot be neglected. Therefore, we decide to choose a random mobility model for users that is realistic and practical. The Smooth Random Mobility describes how the correlation between the speed and the direction is used to provide the smooth movement patterns that are more realistic to be used in the real-life scenarios [9].

Now, with the given mobility model, as discussed in previous section we need to calculate the probability distribution in (8). There are different approaches for predicting the location or trajectory of an individual. The interested reader is referred to the following works, [10], [11] and [12]. Motivated by the work from [11], we partition the spatial area into a grid in which each cell has an area of 25 m^2 and then counts the number of times a mobile user has visited each cell based on the simulation. With this information, we compute a probability distribution representing the likelihood of visiting each particular cell at the time instant t.

Algorithm 1: Localization of mobile users

```
Let us initialize $U$ and $t = 1$ as the user set and
  iteration step, respectively.
**Repeat:** For each $t_c$ seconds:
Search the grid for most probable locations.
Choose from the action set between $n_a = 4$ possible
  locations for each user according to policy derived
  from $Q$ to achieve maximum reward.
**Result:** $xy$-Cartesian coordinates of $U$.
```

A. *Learning based localization*

In this section, we describe the novel technique for localization of mobile users. In the considered scenario, we assume that the initial position of ground users are known to UAV. In our algorithm, with regard to probability distribution found by the grid model, the UAV makes the decision based on the most probable grids which have the highest probabilities. Here, because of the large action size, we limit the choices of UAV at each time instant to $n_a = 4$ for each user. Also, since it is not necessary for the UAV to do the estimation at each time instant, we set a time period T_a in which the UAV will estimate the locations periodically. The localization algorithm is described in the following.

IV. REINFORCEMENT LEARNING FOR TRAJECTORY OPTIMIZATION

In this section, given the location of mobile users, our goal is to obtain the optimal trajectory of the UAV to maximize the system throughput. Reinforcement Learning (RL) has a potential to deal with challenging and realistic models that include stochastic movements of nodes. In general, RL is a learning approach that is used for finding the optimal way of executing a task by letting an entity, named agent, take actions that affect its state within the acting environment. The agent improves over time by incorporating the rewards it had received for its appropriate performance in all episodes [13]. In the Q-learning model, the UAV acts as agent, and the Q-learning model consists of four parts: states, actions, rewards, and Q-value. The aim of Q-learning is for attaining a policy that maximizes the observed rewards over the interaction time of the agent.

1) State Representation: Each state in the set is described as: (x_u, y_u), where (x_u, y_u) is the horizontal position of UAV. As the UAV takes a trajectory in a specific episode, the state space can be defined as $x_u : 0, 1, ...X_d$, $y_u : 0, 1, ...Y_d$, where X_d and Y_d are the maximum coordinate of this particular episode.

Hyperparameter	value
optimizer for SGD	Adam
learning rate for optimizer	0.0001
discount factor γ	0.99
number of hidden layers	2
number of neurons	256
minibatch size	32
action space size	263
activation function	ReLU
replay buffer capacity	10^6

Table I: Training parameters.

2) Action Space: The action space A is described by all possible movement directions, the action of remaining in the same place and 4 possible locations for each of the mobile users. By assuming that the UAV fly with simple coordinate turns, the actions related to movement of UAV is simplified to 7 directions. Combining the actions from the dynamic movement of UAV and estimation based on the grid model, the action size will be equal to 263. More in section IV-A.
3) State Transition Model: Considering a deterministic MDP, there is no randomness in the transitions that follow the agent's decisions. Thus, the next state is only affected by the action that the agent takes.
4) Rewards: The reward function is defined by the instantaneous throughput of users. If the action that the agent carries out at current time t can improve the throughput, then the agent receives a positive reward, otherwise, the agent receives a negative reward.

Due to the size of MDP, we create an RL agent as a feed-forward neural network (NN), with F input neurons, Y hidden states each with the same number of neurons Z, all using rectified linear (ReLU). When receiving the current state, described with F features as input, the NN agent outputs its evaluation for all seven actions that can be taken. However, the use of NNs in RL tasks may fail to converge especially in problems with stochastic environments, such as ours. Therefore, we rely on deep RL and using double Q-learning to solve our problem [14].

For the double-Q-learning RL algorithm, we need to keep two separate agents with the same properties but with different weight values w_P and w_T. As such they will output a different Q-action function when given the same state. One is used to choose the actions, called a primary model $Q_P(s_t, a_t)$, while the other model evaluates the action during the training, called a target model $Q_T(s_t, a_t)$. Therefore training occurs when taking a batch of experiences e_t from the buffer that is used to update the model as:

$$Q_P^{new} = (1 - \alpha)Q_p + \alpha \left[r_t + (1 - d_t)\gamma \max Q_T(s_{t+1}, a)\right] \tag{15}$$

where $\max Q_T(s_{t+1}, a)$ is the action chosen as per the agent, α is the learning rate which was an input to the Adam optimizer [15], and γ is a discount factor that reduces the impact of long term rewards. We implement this with soft updates where instead of waiting several episodes to replace the target

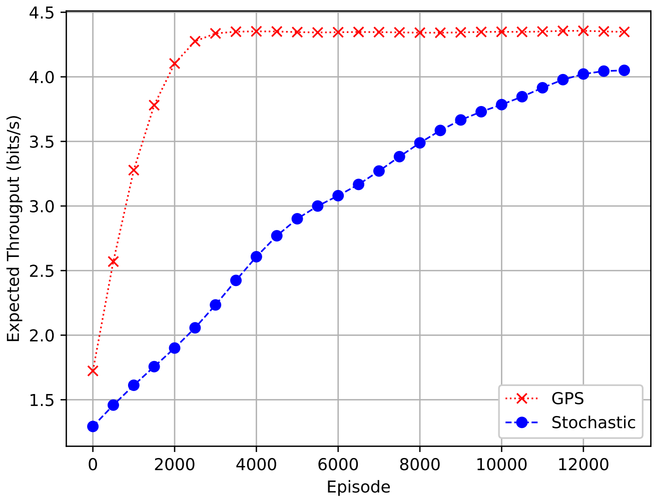


Fig. 2: Convergence of the proposed algorithm vs. the number of training episodes.

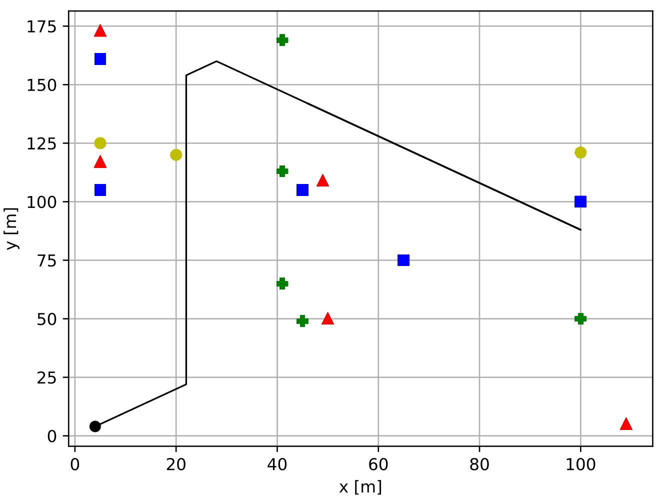


Fig. 3: Trajectory obtained by UAV for the case that four ground users are roaming.

model with the primary. The target model receives continuous updates discounted by value τ as in $w_T = w_T(1-\tau) + w_P\tau$.

A. Dealing with large action space

In this section, we examine how the agent makes the decision from the large action space at each time step and how invalid action masking and normalized probability distribution are realized to strict the agent for repeatedly taking invalid actions. It has been shown that invalid action masking scales better when the space of invalid actions is large and the agent solves the desired task while invalid action penalty struggles to explore even the very first reward.

First, let us see how a normalization carry out in to the discrete action space for when UAV has to decide the location of users after each t_c seconds. For illustration purposes, consider the 4 probabilities in Fig.1 which correspond to highest possible locations for one user at time t. Thus, let us acknowledge an MDP with the action set $A = a_0, a_1, a_2, a_3$ and $S = s, s'$ where the MDP reaches the state s' after an action is taken in the initial state s. Thus we have

$$\begin{aligned} P(s'|s,a) &= [p(a_0|s_0), p(a_1|s_0), p(a_2|s_0), p(a_3|s_0)] \\ &= [0.094, 0.3, 0.104, 0.22] \end{aligned} \tag{16}$$

Now, after normalization enforced, we can write

$$P(s'|s,a) = [0.13, 0.41, 0.14, 0.3] \tag{17}$$

Now for states that UAV actions are about the coordinates of UAV and come from the possible directions described in section IV, we have to mask the invalid actions which correspond to actions related to estimation of user's location. Lets consider our actions space size which is equal to 263. We set the first 7 actions correspond to actions related to direction of UAV and other 256 actions related to user's locations. Suppose that we have an action set $A = a_0, ..., a_6, ..., a_{262}$ in which each action has same probability. Now let us assume that at time instants other than t_c, the actions $[a_7, a_8, a_9, ..., a_{262}]$ are invalid actions and only the first 7 actions are valid. Invalid action masking helps to avoid sampling invalid actions by "masking out" the probabilities corresponding to the invalid actions. This is usually achieved by replacing the probabilities of actions to be masked by zero. Let us use I_a which is stands for this masking process and we can calculate the re-normalized probability distribution $P(s'|s,a)$ as the following:

$$\begin{aligned} P(s'|s,a) &= I_A\left([p_0, ..., p_6, p_7, ..., p_{262}]\right) \\ &= [p'(a_0|s_0), ..., p'(a_6|s_0), p'(a_7|s_0), ..., p'(a_{262}|s_0)] \\ &= [0.142, ..., 0.142, 0, ..., 0] \end{aligned} \tag{18}$$

V. Numerical Results

In this section, we present our numerical results characterising the optimisation problem of UAV-assisted mobile networks. To highlight the efficiency of our proposed model, we compare it to a scenario when UAV is connected to GPS system and has the online knowledge of user's location. We use Tensorflow 2.5.0 and the Adam optimiser for training the neural networks. The training parameters are provided in Table I. In deployment, a 2D area of 1000^2 m is considered. It is assumed that UAV flies at constant altitude and speed H_{uav} = 100m and V_{uav} = 20m/s, respectively. The UAV transmit power is set to P_c = 0.1W and the power of dense noise is assumed to be -174 dB.

In Fig.2, we plot the expected throughput vs the number of training episodes. It can be observed that the UAV is capable of carrying out the actions in an iterative manner and learn from the mistakes for improving the system throughput. In this figure, we also compare our approach to a scenario when the UAV is connected through the GPS system and for the sake of comparison, we assume that the UAV is aware of the user's location at each time instant. As can be seen, the convergence rate of the proposed approach is much slower than the GPS approach. This is due to fact that of the large action space and the stochastic estimation of user's location, which results to necessity of more training episodes.

Fig.3 plots the trajectory of a UAV derived from the proposed approach when ground users move. In this figure, the trajectory of a UAV is shown for the mission duration time of 100 s. In this simulation, we assume that the UAV can move at a constant speed. At each time slot, the UAV choose a direction from the action space which contains 7 directions, then the trajectory will maximize the throughput of ground users. It should be noted that we can adjust the timespan to improve the accuracy of dynamic movement. This, in turn, increases the number of required iterations for convergence. Therefore, a trade-off exists between improving the throughput of ground users and the running complexity of the proposed algorithm.

VI. Conclusion

In this paper, the DRL technique has been utilized to optimize the flight trajectory and throughput performance of UAV-assisted networks. The mobility of users is considered in to the system model and a novel approach for estimating the location of mobile users has been studied. A learning-based algorithm was proposed for solving the problem of maximizing the system throughput by utilising a prior knowledge of likelihood of presence in a grid. We designed a DRL based movement algorithm for obtaining the trajectory of UAV. It is demonstrated that the proposed approach performs well in comparison despite the fact of being simple to implement.

References

[1] J. G. Andrews, S. Buzzi, W. Choi, S. V. Hanly, A. Lozano, A. C. Soong, and J. C. Zhang, "What will 5g be?," *IEEE Journal on selected areas in communications*, vol. 32, no. 6, pp. 1065–1082, 2014.

[2] V. W. Wong, R. Schober, D. W. K. Ng, and L.-C. Wang, *Key technologies for 5G wireless systems*. Cambridge university press, 2017.

[3] I. Valiulahi and C. Masouros, "Multi-uav deployment for throughput maximization in the presence of co-channel interference," *IEEE Internet of Things Journal*, vol. 8, no. 5, pp. 3605–3618, 2020.

[4] A. Merwaday and I. Guvenc, "Uav assisted heterogeneous networks for public safety communications," in *2015 IEEE wireless communications and networking conference workshops (WCNCW)*, pp. 329–334, IEEE, 2015.

[5] M. Mozaffari, W. Saad, M. Bennis, and M. Debbah, "Unmanned aerial vehicle with underlaid device-to-device communications: Performance and tradeoffs," *IEEE Transactions on Wireless Communications*, vol. 15, no. 6, pp. 3949–3963, 2016.

[6] M. Zhang, S. Fu, and Q. Fan, "Joint 3d deployment and power allocation for uav-bs: A deep reinforcement learning approach," *IEEE Wireless Communications Letters*, 2021.

[7] C. Zhao, J. Liu, M. Sheng, W. Teng, Y. Zheng, and J. Li, "Multi-uav trajectory planning for energy-efficient content coverage: A decentralized learning-based approach," *IEEE Journal on Selected Areas in Communications*, 2021.

[8] 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, 2019.

[9] C. Bettstetter, "Smooth is better than sharp: A random mobility model for simulation of wireless networks," in *Proceedings of the 4th ACM international workshop on Modeling, analysis and simulation of wireless and mobile systems*, pp. 19–27, 2001.

[10] A. Asahara, K. Maruyama, A. Sato, and K. Seto, "Pedestrian-movement prediction based on mixed markov-chain model," in *Proceedings of the 19th ACM SIGSPATIAL international conference on advances in geographic information systems*, pp. 25–33, 2011.

[11] J. Krumm and E. Horvitz, "Predestination: Inferring destinations from partial trajectories," in *International Conference on Ubiquitous Computing*, pp. 243–260, Springer, 2006.

[12] J. J.-C. Ying, W.-C. Lee, T.-C. Weng, and V. S. Tseng, "Semantic trajectory mining for location prediction," in *Proceedings of the 19th ACM SIGSPATIAL international conference on advances in geographic information systems*, pp. 34–43, 2011.

[13] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.

[14] H. Hasselt, "Double q-learning," *Advances in neural information processing systems*, vol. 23, pp. 2613–2621, 2010.

[15] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.