Meta-Transfer Learning-Based Handover Optimization for V2N Communication

Rana Muhammad Sohaib, *Member, IEEE,* Oluwakayode Onireti, *Senior Member, IEEE,* Kang Tan, *Member, IEEE,* Yusuf Sambo, *Senior Member, IEEE,* Rafiq Swash, *Member, IEEE.* and Muhammad Imran, *Fellow, IEEE,*

***Abstract*—The rapid growth of vehicle-to-network (V2N) communication demands efficient handover decision-making strategies to ensure seamless connectivity and maximum throughput. However, the dynamic nature of V2N scenarios poses challenges for traditional handover algorithms. To address this, we propose a deep reinforcement learning (DRL)-based approach for optimizing handover decisions in dynamic V2N communication. We leverages the advantages of transfer learning and meta-learning to generalize across time-evolving source and target tasks. In this paper, we derive generalization bounds for our DRL-based approach, specifically focusing on optimizing the handover process in V2N communication. The derived bounds provide theoretical guarantees on the expected generalization error of the learned handover time function for the target task. To implement our framework, we propose a meta-learning framework, Adapt-to-evolve (A2E), based on the double deep Q-networks (DDQN) with Thompson sampling approach. The A2E framework enables quick adaptation to new tasks by minimizing the error upper bounds with divergence measures. Through transfer learning, the meta-learner dynamically evolves its handover decision-making strategy to maximize average throughput while reducing the number of handovers. We use Thompson sampling with the DDQN to balance exploration and exploitation. The DDQN with The Thompson sampling approach, ensuring efficient and effective learning, forms the foundation for optimizing the meta-training process, resulting in improvement in cumulated packet loss by 48.02 % in highway settings and 46.32 % in rural settings.**

***Index Terms*—V2N, DRL, HO, generalization, meta-learning.**

I. Introduction

INTELLIGENT transportation systems (ITS) are emerging as an essential element to enhance daily existence, tackling the fundamental objective of enhancing on-road security and vehicular gridlock while offering diverse utility-oriented on-board amenities [1], [2]. Internet-of-Vehicles (IoV) networks plays a crucial role in facilitating data exchange within the ITS domain. The purpose of IoV networks is to guarantee roadway security, enhance transportation effectiveness, and offer a fresh degree of on-vehicle amusement. To accomplish these objectives, a vehicle must establish communication with any vehicle and entity that may affect or may be affected by the vehicle, and this is generally termed vehicle-to-everything (V2X) communication [3]. Diverse categories of interaction exist within V2X networks depending on the entity with which a vehicle establishes a connection, encompassing vehicle-to-vehicle (V2V), vehicle-to-pedestrian (V2P), vehicle-to-network (V2N) and vehicle-to-infrastructure (V2I) [4]. Evaluating quality-of-service (QoS) within these networks involves assessing critical technological aspects like reliability, scalability, and network congestion, measured through metrics such as throughput, packet loss, error rates, and latency [5]. The integration of V2N communications is set to have a pivotal role in enabling the advancement of forthcoming vehicular networks [6]. This integration will enable the delivery of on-board infotainment services by ensuring good connectivity to the network. This, in turn, will necessitate a significant data transfer rate between vehicles and base stations (BSs), which parallels the operational requirements of conventional cellular network user equipments (UEs). In wireless networks, the movement of UEs is managed through the handover (HO) process, which transfers ongoing communication sessions from one BS to another, allowing UEs to transition between coverage regions of various BSs seamlessly while ensuring uninterrupted sessions. The significantly growing numbers of BSs and links which necessitates additional HO operations will result in exceedingly intricate mobility management. Authors in [7] propose two traffic-aware spectrum handover schemes for cognitive radio heterogeneous networks to enhance spectrum utilization and ensure quality of service for licensed primary users. These schemes are designed to operate in both distributed and centralized manners, balancing performance and complexity. However, traditional HO algorithms often struggle to cope with the dynamic and evolving nature of V2N communication, necessitating innovative approaches to optimize the HO process.

The utilization of machine learning (ML)-based methods can make a substantial impact on handover optimization by reducing latency, overhead, and frequent handovers [8]. Deep reinforcement learning (DRL) has emerged as a promising approach for addressing complex decision-making problems [9]. By leveraging neural networks and reinforcement learning algorithms, DRL enables agents to learn optimal policies directly from raw input data. Artificial Neural Network (ANN) empowered DRL has achieved notable progress in domains characterized by complexities and fluctuations [10], [11]. It stands as a propitious approach for devising efficient remedies to overcome the challenges associated with HO and has garnered substantial research impetus. Thus far, a substantial amount of investigation has been conducted regarding the subject matter of HO optimization using ML, employing diverse sets of input variables and incorporating

R. M. Sohaib, O. Onireti, K. Tan, Y. Sambo and M. A. Imran are with the James Watt School of Engineering, University of Glasgow, Glasgow G12 8QQ, UK e-mail: (2500800s@student.gla.ac.uk, k.tan.3@research.gla.ac.uk, {Oluwakayode.Onireti, Yusuf.Sambo, Muhammad.Imran}@glasgow.ac.uk). Rafiq Swash is with the AIDRIVERS LTD. E-mail: swash@aidrivers.ai

Manuscript received April 19, 2005; revised August 26, 2015.

This article has been accepted for publication in IEEE Transactions on Vehicular Technology. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TVT.2024.3431875

a range of network architecture configurations [12], [13]. Nevertheless, a significant portion of the prevailing literature primarily concentrates on particular application scenarios and the design of system architectures, rarely delving into the exploration of real-world implementation scenarios [14]. In recent times, ML-based approaches have been extensively investigated within diverse studies related to wireless communication, covering resource allocation, power control, and HO management across multiple platforms [15]. ML algorithms have the capability to harness the abundant dataset produced by wireless systems and uncover concealed patterns within the data that typically prove challenging to extract using analytical optimization methods [16]. Different studies have been carried out to enhance the effectiveness of triggering and decision-making in HO [12]. There are three primary categories of ML approaches for optimizing HO: ML-driven parameter optimization for conventional HO, ML-powered direct decision-making for HO, and ML-assisted optimization for HO [17]. An algorithm utilizing Q-learning, presented by [18], was suggested to enhance the HO parameters. By configuring the reward function to encompass system-wide factors, the proposed approach effectively optimized the values associated with time-to-trigger (TTT) and hysteresis. An instance of employing machine learning can be found in the research conducted in [19], where a novel ML technique was introduced and investigated. This approach aimed to ascertain optimal timing and positioning for HOs within 5G radio networks, and also to determine how the acquired model could be utilized to initiate HOs based on predicted radio conditions. Likewise, the study conducted by authors in [20] employed Q-learning in conjunction with an analytic hierarchy process technique for prioritizing similarities to an ideal solution. This approach was employed to enhance two parameters of higher-order systems, namely hysteresis and TTT.

For the purpose of ML-driven HO decision-making, authors in [21] presented a K-means clustering technique, with the objective of grouping UEs according to their movement patterns. Subsequently, an asynchronous multi-agent DRL algorithm was employed to achieve optimal HO decisions. In [22], the researchers introduced a novel approach utilizing Reinforcement Learning (RL) to establish a framework for managing HO in heterogeneous networks (Het-Nets). They focused on collectively acquiring knowledge on traffic load and determining the optimal expansion range of both macro and small cells. Furthermore, users were prioritized based on their velocities and past HO rates to improve the overall user throughput. The work in [23] employs a two-tier ML-driven model for the management of HO in vehicular networks. The first tier of the model utilized Recurrent Neural Networks (RNN) to forecast the Received Signal Strength (RSS) required for HO activation. Subsequently, a stochastic Markov model was employed in the second tier to determine the selection of BS for HO. In [24], a novel approach was devised to create a cohesive HO algorithm for LTE-A systems. The foundation of this algorithm relied on discrete stochastic dynamic programming, taking into account the combined factors of UE measurements such as reference signal received power (RSRP) and reference signal received quality (RSRQ), along with the holistic assessment of resource utilization. The outcome of this methodology was a set of HO decisions that effectively achieved load balance. Authors in [25] employed simulated signal-to-interference-and-noise ratio (SINR) maps and the deep Q-learning technique to determine dynamic HO in a vehicular network. Their research utilized event A2 to initiate HOs, leveraging its ability to detect potential obstructions while simultaneously expediting the training of ANNs by bypassing unimportant states. Moreover, an integrated HO and power distribution strategy was formulated for Het-Nets employing multi-agent DRL [26]. The algorithm enhanced the selection of BSs and power levels for every UE by employing a reward scheme that relied on system efficiency and penalty for HO. Authors in [27] investigated a distributed Q-learning approach to address the challenges associated with HO in the context of network slicing. The goal was to enable a UE to determine whether a HO was necessary within a network slicing configuration. The work in [28] introduces a novel federated learning (FL) training framework aimed at predicting signal-to-noise ratio (SNR). This innovative approach seamlessly integrates both the macro BS and the dynamic local UEs. The traditional HO algorithm was enhanced with SNR predictions, which allowed for dynamic HOs in a vehicular network. Using advanced vehicle trajectory predictions aligned with established BS locations, the approach presented in [29] possesses the ability to preemptively initiate optimal HOs, thereby minimizing the complexities of HO decision-making. Hybrid ML-based HO schemes also exist in addition to the above types. In [30], an approach was devised for optimizing HO by employing a combination of a RNN and a multi-layer perception neural network. These networks expertly harnessed diverse information collected across the LTE protocol stack, and contributing invaluable assistance to the decision-making process for optimal HO. Authors in [31] proposed a long short-term memory (LSTM)-based RNN approach to forecast subsequent received signal strength indication (RSSI) for proactive HO triggering. Subsequently, a Hidden Markov Model [32] was employed to improve the HO decision-making procedure.

The literature discussed typical cellular networks with slow-moving UEs, but scenarios involving vehicular UEs moving at fast speeds and stringent QoS demands were very few. However, only a few research has examined how well ML-based solutions perform compared to conventional methods using the same information. Furthermore, only a few studies have explored the performance of various ML-based solutions using standardized datasets and test environments [33],[17]. Out of those, just [17], [24], [30], and [31] used a full-stack simulator (e.g., ns-2 and ns-3) to assess and compare the performance of their proposed schemes. According to the author's knowledge, no investigation has been carried out on how the ML-based HO optimization will perform when a different set of information is fed to the model compared to the training data. Existing literature considers a similar set of information for the duration of training and testing. However, this assumption is unrealistic in the context of a real-time V2X environment scenario. Existing studies lack the ability to adapt to various environments. The existing DRL-centric methodologies are formulated under the premise

This article has been accepted for publication in IEEE Transactions on Vehicular Technology. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TVT.2024.3431875

that there exists a consistent and unchanging environment for both training and testing. Nevertheless, this assumption proves to be unfeasible within V2X communication contexts, given the substantial mobility and ever-changing attributes exhibited by vehicular surroundings. Consequently, it can give rise to a disparity predicament as the environment changes. This implies that such algorithms are incapable of promptly making accurate decisions in a fast varying channel environment. These challenges remain unresolved and impede the progress of effective HO methodologies in V2X communications.

In this paper, we aim to tackle the fast-changing channel conditions in the V2N HO scenarios by combining transfer learning and meta-learning with DRL. We create practical simulation scenarios of a V2N communication using the full-stack ns-3 simulator. Using transfer learning and meta-learning can tackle challenges like limited computational resources and generalization. Thus, making them more practical, scalable and crucial for unlocking their full potential in future transportation and smart city applications. To address these challenges, this paper proposes a novel framework for optimizing V2N communication HO using a DRL-based approach. The key objective of our research is to develop an intelligent HO decision-making mechanism that can adapt to the dynamic V2N environment, maximize average throughput, and minimize the number of HOs. Achieving this requires leveraging the capabilities of transfer learning and meta-learning to generalize across time-evolving source and target tasks. Our framework enables the agent to learn from historical data, transfer knowledge across tasks, and dynamically evolve its HO decision-making strategy to optimize network performance. Incorporating transfer learning and meta-learning in V2N communication handover addresses challenges by leveraging prior knowledge, enhancing adaptability, facilitating rapid deployment and scalability, improving robustness and generalization, and optimizing resource utilization. The proposed approach exclusively employed the RSRP as the input. The dataset used for training was directly collected from the cellular protocol stack's network layer through simulation. The cellular network module provided by ns-3 was employed in conformity with the established guidelines set forth by the 3rd Generation Partnership Project (3GPP) [34]. The main contributions of this work are as follows:

- We introduce a comprehensive framework that addresses the challenges of dynamic HO decision-making in V2N communication. The framework combines transfer learning, meta-learning, and the double deep Q-networks (DDQN) with the Thompson sampling (TS) approach to enable intelligent HO optimization.
- We derive generalization bounds specific to the HO optimization problem in dynamic V2N communication. These bounds provide theoretical guarantees on the expected generalization error of the learned HO time function for the target task.
- A meta-transfer learning framework adapt-to-evolve (A2E) based on the DDQN with TS algorithm is proposed to facilitate efficient and effective HO optimization. The meta-learner quickly adapts to new tasks by minimizing the error's upper bound. This allows the agent to optimize HO decisions while considering the specific characteristics of each task, leading to improved performance and generalization capabilities.
- We conduct extensive experiments using different realistic V2N communication scenarios to evaluate the performance of our proposed approach. We compare our framework against traditional HO algorithms and state-of-the-art approaches in terms of average throughput, SINR improvement, and HO frequency.

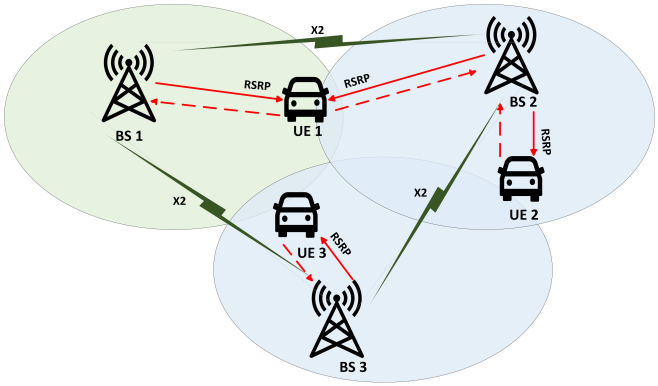

Fig. 1: System model of V2N communication

The remaining portion of this paper is structured as follows: The system model and problem statement are presented in Section II; the DRL-based solution is specified in Section III, followed by the generalization bound and the proposed framework in Section IV and V, respectively. Section VI presents the simulation results. Finally, the conclusion and future research insight are presented in Section VII.

II. System Model & Problem Formulation

In this work, we explore a cellular V2N network design, comprising vehicular UEs (VUEs) and BSs, aiming to optimize the HO efficiency. Fig. 1 depicts the V2N scenario where only VUEs are connected with the BSs. In cellular networks, each VUE measures and reports two important values, RSRP and RSRQ, to determine the strength and quality of the connection to the BSs. These measurements help in making decisions on HO connections between BSs. Following the 3GPP standard [35], RSRP represents the combined power of resource elements from BS-specific reference signals within the given bandwidth. RSRQ, on the other hand, encompasses channel interference and thermal noise as well. The following equation shows how RSRP and RSRQ are related:

$$RSRQ = \wp \times \frac{RSRP}{RSSI}, \tag{1}$$

where $\wp$ represents the number of resource blocks (RBs). HO decision-making relies on VUE measurement reports, with a particular focus on events A2 and A3 for intra-radio access technology HO initialization [36]. Event A2 occurs when the measured RSRP or RSRQ drops below a certain threshold. On the other hand, event A3 occurs when the neighboring BS RSRP/RSRQ is above that of the serving BS by a predefined margin. Cellular networks use two distinct and standardized

This article has been accepted for publication in IEEE Transactions on Vehicular Technology. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TVT.2024.3431875

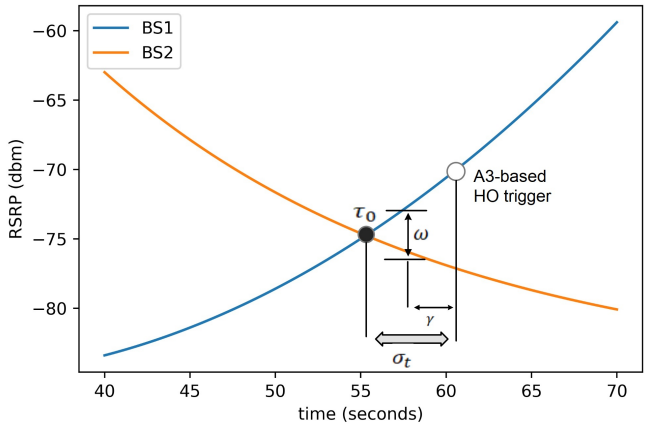

Fig. 2: A3-based HO trigger, where, τ_0 depicts the optimal HO triggering instant, ω and γ specify the hysteresis and TTT, respectively, while σ_t is the time delay. Note that the figure is generated via a simple simulation, with on X and Y axis only reflecting the values obtained by the simulation.

HO algorithms namely A2A4 and A3 to trigger and make HO decisions [37]. In Fig. 2, we show the A3-based HO approach, which uses two HO control parameters, hysteresis ω and TTT γ [38]. The hysteresis ω ensures the target cell's RSRP exceeds the serving cell's RSRP by a margin, reducing unnecessary handovers.The TTT γ imposes a waiting period to consistently meet this condition, minimizing ping-pong effects and enhancing network stability.

A. Handover Delay Cost

Existing cellular networks use hard HO, which cuts the connection between the device and its current BS before establishing a new one [39]. During hard HO, choosing the best BS becomes crucial because the UE won't receive any meaningful information about services during the HO. The HO delay time t_d refers to the time taken by the UE to switch connections from its serving BS to the target BS after the HO trigger. The gradual aggregation of HO delay time causes the UE's average throughput to degrade. HO cost β_c refers to the result of cumulative HO delay time and it can be expressed as the product of t_d and total number of HO $\mathcal{M}_{ho}$ in a specific trajectory [40] such that:

$$\beta_c = \mathcal{M}_{ho} \times t_d. \tag{2}$$

The normalized HO delay cost ζ_c of a VUE for a specific time period T can be expressed as:

$$\zeta_c = \min\left(\frac{\beta_c}{T}, 1\right). \tag{3}$$

The parameter ζ_c ranges between 0 and 1, and it represents the proportion of time invested in the HO procedure. As ζ_c approaches 1, it implies that the UE has used nearly the entire time span T on HOs. Consequently using Shannon's equation, the average throughput can be computed as:

$$\Omega = \mathcal{B} \times (1 - \zeta_c) \times \log_2(1 + \aleph) \tag{4}$$

where $\mathcal{B}$ and $\aleph$ refers to the bandwidth and SINR, respectively. Hence, given a constant bandwidth B and HO delay time t_d, the number of HOs $\mathcal{M}_{ho}$ and SINR $\aleph$, play an important role in achieving high throughput in a specific trajectory and transverse time. Consequently, we look to maximize system throughput by making the HO decision process optimal while avoiding the frequent HOs. The optimization problem can be formulated as follows

$$\arg\max_{\mathcal{M}_{ho}} \Omega \tag{5a}$$

$$\text{s.t. } \zeta_c \leq 1, \quad \zeta_c \in [0, 1] \tag{5b}$$

III. DDQN with Thompson Sampling

In this section, we present a DRL-based approach to intelligently optimize the HO decision-making with the aim of maximizing the throughput of VUE in a given trajectory. In RL, an agent takes decisions according to the present observed environment, future state observations, and rewards. Although traditional RL approaches work well in simple settings, they struggle to cope with highly complex outdoor radio environments, where optimal decision-making for HO tasks is required due to the large amount of data involved. Consequently, alternative approaches emerged to address this, such as linear value function approximation and policy gradient methods [41]. Deep learning (DL) techniques, employing ANN and DRL approaches, led to the creation of a powerful method known as Deep Q-network (DQN) [42]. DQN combines Q-learning, a model-free RL algorithm, with ANN to excel in training RL tasks in challenging environments where traditional RL struggles. In RL training, the Bellman equation [41] holds the key, the Q-value in DQN is calculated as follows:

$$Q(s_t, a_t; \theta) = r(s_t, a_t) + \Lambda Q\left(s'_t, \arg\max_{a'_t} Q(s_t, a'_t; \theta); \theta\right), \tag{6}$$

where s_t and a_t refer to the state space and action space, respectively. Furthermore, r and Λ are the received reward and the discount factor (with values ranging from 0 to 1). The parameter θ represents the Q-table's approximation in the form of a ANN. Using a single network, DQN calculates the prediction Q value $Q(s', a')$ and the updated Q value $Q(s, a)$. However, the shared parameter θ can cause instability during training, leading to potential non-convergence as the current $Q(s, a; \theta)$ alters the values of future states when its parameters are updated. In [43], authors utilized DDQN to address the overestimation of action values by using two separate neural networks to independently estimate the action values. One network is used to select the best action for the next state in the target network θ', while the other network estimates the Q-values in the current state. By decoupling the target and online networks, DDQN reduces the overestimation bias and stabilizes the learning process, resulting in improved performance. During DDQN training, θ is continually updated, whereas θ' periodically aligns with θ by adopting every parameter to maintain up-to-date information. Hence, (6) can be updated as:

$$Q(s_t, a_t; \theta) = r + \Lambda Q\left(s'_t, \arg\max_{a'_t} Q(s'_t, a'_t; \theta); \theta'\right). \tag{7}$$

DQN and the target network are the two different deep neural networks utilized by DDQN. The DDQN can be expressed as:

$$y \leftarrow r_{t+1} + \Lambda \hat{Q}(s_{t+1}, \hat{a}), \tag{8}$$

where

$$\hat{a} = \max_{a} Q_{DQN}(s_{t+1}, a) \tag{9}$$

and $\hat{Q}(s_t + 1, \hat{a})$ denotes the target network. The loss function can be expressed as:

$$\nabla(Q, \hat{Q}) = \mathbb{E}\left[(Q(s_t, a_t) - \hat{Q}(s_t, a_t))^2\right]. \tag{10}$$

In this work, we aim to develop and enhanced HO approach for V2N communications and compare its performance with the simple DDQN approach [17]. The centralized agent is trained on RSRP values collected from the V2N communication network. During training, the agent learns to estimate the expected rewards of different HO decisions, considering the uncertainty in the environment. The HO problem is modeled as a Markov Decision Process (MDP), defined by the tuple (s, a_t, P, r), where s and a_t are the state and action spaces defined above, P is the transition probability, and r is the reward function.

1) State space: Researchers have extensively studied strategies for selecting BSs in vehicular networks based on mobility by using a user's location and speed [12]. Instead of directly measuring a user's location, it is more practical to estimate the RSRP information [44]. A clear connection is established between a specific geographic spot in a defined region and a collection of RSRP values from the BSs within that area. So, this research takes into account the combination of RSRP values measured by a VUE from all nearby BSs. We assume that all VUEs are at the same height. For a number of BSs k and a VUE located at location l in a given trajectory, the RSRP measurements Γ_l can be expressed as

$$\Gamma_l = \left\{rsrp_l^1, rsrp_l^2, ..., rsrp_l^k\right\}. \tag{11}$$

Thus, the state space vector s_l contains the RSRP and serving BS identifier (ID).We use the one-hot encoding [45] method to illustrate the state space. Let's assume that if there are 4 BSs in the given trajectory and a VUE's serving BS's ID is 3, then the serving BS ID is denoted by $\{0, 0, 1, 0\}$. So, the state space can be expressed as:

$$s_l = \left\{\Gamma_l; BS_{id}\right\}. \tag{12}$$

We assume that the VUE observes and reports the environment at regular intervals of time during training and evaluation.

2) Action space: An action space can be described as the process of connecting a VUE to the next state which involves choosing a BS from the ones listed in the given trajectory, including the serving BS when necessary. If the action chosen instructs the device to connect to a nearby BS, HO will occur. However, if the action indicates the current serving BS, the device will stay connected without the need for HO. Thus, the action space can be expressed as:

$$a_t = \left\{BS_0, BS_1, ..., BS_k\right\}, \tag{13}$$

where vector comprises the IDs of local BSs.

3) Reward: The purpose of the reward function is to encourage the agent to take actions that will maximize the overall reward over time. Our objective is to attain the highest system throughput Ω by reducing the ζ_c, as highlighted in (4). To achieve this, the number of HOs and time delay should be minimized. The number of HOs can be managed by executing the HO skipping policy. Usually, the agent starts indirect TTT without a fixed value,[1] but it must be done smartly such that VUE attains maximum throughput, even if some HO steps are skipped. Furthermore, during HO the centralized agent may choose BS that offers less numbers of HO events in the future. Hence, we design the reward, which measures the impact of an action in reaching the agent's objective. The reward function can be expressed as:

$$r(s_l, a; s_{l+1}) = \begin{cases} \mathcal{B} \times (1 - \zeta_c) \times \log_2(1 + \aleph), \\ \text{if HO happens} \\ \\ \mathcal{B} \times \log_2(1 + \aleph), \\ \text{otherwise.} \end{cases} \tag{14}$$

A. Thompson Sampling

TS is a method for online decision-making under uncertainty [46]. It is used for balancing exploration and exploitation in reinforcement learning. In this work, we introduce TS-based DDQN centralized agent to optimize HO decisions in cellular networks. Instead of maintaining Q-values or action probabilities, TS uses probability distributions to model uncertainty about the true values of actions. The algorithm then samples from these distributions and selects actions based on the sampled values. This stochastic approach allows the agent to explore different actions while favoring actions that have shown promise in the past. In HO decisions, the network may face uncertain conditions such as changing wireless channel conditions, varying user demands, or mobility patterns. TS can effectively handle uncertainty by incorporating a probabilistic approach. This makes it well-suited for scenarios where the quality of the communication link is subject to fluctuations and uncertainties. Its probabilistic nature encourages the agent to explore different HO decisions, ensuring that it doesn't get stuck in suboptimal choices. At the same time, DDQN provides a mechanism to exploit the knowledge gained from the DNN, helping the agent make informed decisions based on historical data. In the exploration-exploitation trade-off, TS selects actions based on their probability of being optimal. It maintains a posterior distribution over the Q-values, and samples from this distribution to select actions. TS can be applied by modeling the rewards of different actions in a probabilistic manner. Instead of directly selecting the action with the highest estimated reward (as done in traditional Q-learning), TS randomly samples from the posterior distribution of the rewards and selects the action associated with the highest sample. In this work, we use a Gaussian distribution as the posterior, with the mean and variance estimated by maintaining a probability distribution for each action's expected reward.

[1]Indirect TTT refers to a method where the TTT value is not fixed but rather determined indirectly based on various network and device conditions.

We use Bayesian linear regression (BLR) to handle the distribution over Q-values. BLR can provide a posterior distribution over the Q-values given a set of observations. The Q-value of each state-action pair is modeled as a linear function of the state-action features with additive Gaussian noise, and the model parameters (i.e., the coefficients of the linear function) have a Gaussian prior. By observing a set of transitions, we can update the posterior of the model parameters using the Bayesian rule, and hence obtain the posterior distribution over the Q-values. Let's assume the Q-value of a state-action pair is a linear function of a set of features $\phi(s,a)$ extracted from the state-action pair, i.e., $Q(s,a) = \phi(s,a)^T w$, where w is a vector of model parameters, and $\phi(s,a)$ is a feature vector. Given a set of observed transitions (s,a,r,s'), the target of the Q-value update is $y = r + \Lambda * \max'_a Q(s',a')$, and we have $y = \phi(s,a)^T w + \epsilon$, where ϵ is an additive Gaussian noise with zero mean and variance υ^2. The prior of w is assumed to be a Gaussian distribution $N(w|0, \Psi_{prior})$, where Ψ_{prior} is the prior covariance matrix, typically chosen as a scaled identity matrix. Given a set of observations $D = (s,a,r,s')$, the posterior of w is also a Gaussian distribution $N(w|\mu_{post}, \Psi_{post})$, where the mean μ_{post} and covariance Ψ_{post} can be computed by using the Bayesian as

$$\Psi_{post} = \left(\Psi_{prior}^{-1} + \frac{1}{\upsilon^2} \phi^T \phi \right)^{-1} \tag{15}$$

and

$$\mu_{post} = \Psi_{post} \left(\frac{1}{\upsilon^2} \phi^T Y \right), \tag{16}$$

respectively, where ϕ is a matrix whose rows are the feature vectors $\phi(s,a)$ of the observed transitions, and Y is a vector of the targets y of the observed transitions. The posterior distribution of the Q-value $Q(s,a) = \phi(s,a)^T w$ is therefore a Gaussian distribution $N(Q(s,a)|\mu_{s,a}, \upsilon^2_{s,a})$, where $\mu_{s,a} = \phi(s,a)^T \mu_{post}$ and $\upsilon^2_{s,a} = \phi(s,a)^T \Psi_{post} \phi(s,a)$. In TS, when we are in state s and need to select an action, we sample a Q-value from the posterior distribution $N(Q(s,a)|\mu_{s,a}, \upsilon^2_{s,a})$ for each action and choose the action with the maximum sampled Q-value. This allows us to capture the uncertainty over the Q-value estimates and make efficient exploration-exploitation. Algorithm 1 provides the training details.

Algorithm 1 DDQN with Thompson sampling

1: Initialize the replay buffer capacity
2: Initialize the Gaussian BLR model with prior mean 0 and prior covariance
3: Initialize θ, and θ'
4: **for** each episode **do**
5: Observe the network state $s_t = \{RSRP, BS_{ID}\}$
6: Select an action according to TS: sample a Q-value from the posterior distribution $N(Q(s,a)|\mu_{sa,}, \upsilon^2_{s,a})$ for each action, and choose the action with the maximum sampled Q-value.
7: Take this action and then observe the reward and the next state.
8: Store transition (s_t, a_t, r_t, s_{t+1}) in replay memory
9: **if** replay memory is full **then**
10: Sample a mini-batch from the replay memory
11: Compute the target $y = r + \Lambda * \max'_a Q(s',a')$ for each transition in the mini-batch, and update the posterior of the model parameters using the Bayesian rule.
12: **end if**
13: Every 7 steps, update the target network by copying the weights from the online network.
14: **end for**

IV. Generalization bound

In the previous section, we present an intelligent DRL-based approach to address the HO problem in a stationary environment. The TS-based DDQN framework considers the same environment for the duration of training and testing. However, this assumption is non-ideal in the context of a real-time V2N HO scenario, where the information changes rapidly. Existing research does not highlight the generalization aspects for different environment scenarios because there will be a discrepancy if the testing data has a different distribution compared to the training data. In this section, we derive generalization bounds specific to the HO optimization problem in a dynamic V2N communication scenario. The error upper bounds provide a measure of how well the model is expected to perform on the target task based on its performance on the source tasks. By minimizing the error upper bounds, we can ensure that the model is well-suited for the target task during the HO, while also leveraging the knowledge that it has acquired from the source tasks. The error bounds theorem provides a theoretical guarantee that the expected error between the predicted optimal HO time and the true optimal HO time can be bounded by a constant term and a complexity term that depends on the distance measure between the source tasks and the target task. To measure the discrepancies between all tasks we utilize $\mathcal{H}$-divergence [47].

Let us assume that $\mathbb{D}^s$ and $\{\mathbb{D}^t_k\}_{k=1}^K$ refer to the static source task and dynamic target task at time slot k, respectively. Let z^s denote the total number of labeled training samples in the source task, which can be represented as $D^s = \{(a^s_j, b^s_j)\}_{j=1}^{z^s}$. We assume that there are z^t_k unlabeled target task training samples, $D^t_k = \{a^t_{jk}\}_{j=1}^{z^t_k}$. In this work, we look to learn the estimation function for the newest target task $\{\mathbb{D}^t_{N+1}\}$ by capitalizing on the past source and target task. We define the expected error on source task as $E^s(h) = \mathbb{E}_{(a,b)\sim D^s}[L(h(a),b)], \forall h \in \mathcal{H}$, where L refers to the loss function. The empirical error can be defined as $\hat{E}^s(h) = \frac{1}{z^s} \sum_{j=1}^{z^s} L(h(a_j), b_j)$. The following theorem presents the error bounds for dynamic transfer learning with time-evolving source and target tasks in the context of HO.

Theorem 1. *Let $\mathcal{H}$ be the hypothesis class of all possible HO time functions, such that $h : a \to b$. Here, a is the HO time function learned by Algorithm 1 for the target task based on the source domain task. And let b be the optimal HO time for the target task. We assume that at each time slot k we have z number of labeled source samples from the source task $\mathbb{D}^s$ (can be represented as $\mathbb{D}^t_0$) and z labeled target samples from the target task. Given a certain learning rate β and probability*

threshold $\delta > 0$, the expected error of the newest target task can be bounded as

$$E_k^t(h) \leq \sum_{j=0}^{K} \sum_{k=j+1}^{K+1} \nu_{jk} \left(\hat{E}_j^t(h_j) + \varrho_{jk}.\hat{d}_{\mathcal{H}\Delta\mathcal{H}}(\mathbb{D}_j^t, \mathbb{D}_k^t) \right) + \mathcal{O}\left[\sum_{j=0}^{K} \left(\frac{1}{z} \sum_{i=1}^{z} ||\nabla_\theta \bar{h}(a_{ij})|| \right)^2 + \xi + \sqrt{\frac{\log(2z) + \log(2/\delta) + \sum_{j=0}^{K} \sum_{k=j+1}^{K+1} \nu_{jk}^2 \log(1/\delta)}{z}} \right] \tag{17}$$

where ξ refers to the total error over all the tasks, i.e., $\xi = \min_{h \in H} \sum_{j=0}^{K+1} E_j^t(h)$, and $\hat{d}_{\mathcal{H}\Delta\mathcal{H}}(.,.)$ indicates the empirical estimate. Parameter ν_{jk} refers to the sampling probability which plays an essential part in the generalization error bound. ϱ_{jk} indicates the hyper-parameter which helps to achieve a balance between reducing classification errors and discrepancies.

Proof: See the Appendix □

V. Adapt-to-evolve (A2E) Framework

In this section, we present a meta-transfer learning-based approach named ADAPT-TO-EVOLVE (A2E) and look to minimize the error bound derived in the previous section. Theorem 1 demonstrates that we can limit the expected error of the newest target task by considering the past source and target information. We develop a method to automatically create meta-tasks from the dynamic target task. Our aim is to dynamically learn the sampling probability parameter ν which is associated with the classification error on the target task. The objective function to learn the estimation function of $\mathbb{D}_{K+1}^t$ at time $K+1$ can be formulated as:

$$\min_{\theta} \min_{\nu} J(\theta, \nu) = \sum_{j=0}^{K} \sum_{k=j+1}^{K+1} \nu_{jk} \left(\hat{E}_j^t(\mathcal{F}_{jk}(\theta)) + \varrho.\hat{d}_{\mathcal{H}\Delta\mathcal{H}}\left(\mathbb{D}_j^t, \mathbb{D}_k^t; \mathcal{F}_{jk}(\theta) \right) \right) \tag{18a}$$

$$\text{s.t.} \quad \sum_{j=0}^{K} \sum_{k=j+1}^{K+1} \nu_{jk} = 1 \tag{18b}$$

$$\text{s.t.} \quad \mathcal{F}_{jk}(\theta)) = \theta - \mho \nabla_\theta L\left(\mathbb{D}_j^t, \mathbb{D}_k^t \right) \tag{18c}$$

where $L\left(\mathbb{D}_j^t, \mathbb{D}_k^t\right)$ indicates the meta-training loss and θ refers to the trainable parameter. $\mho$ and ∇ refer to the learning rate and gradient over weight parameters, respectively, and $\mathcal{F}_{jk}(\theta)$ refers to the mapping function. (18b) refers to the sampling probability constraint, whereas (18c) ensures that the mapping function $\mathcal{F}_{jk}(\theta)$ should be updated in a way that is consistent with gradient descent. This helps in maintaining the consistency and stability of the optimization process while learning the sampling probability parameter ν and estimating $\mathbb{D}_{K+1}^t$ at time $K+1$.

A. Meta-tasks

Theorem 1 states that parameter ν_{jk} depends significantly on the classification error on the target task and the difference in empirical distributions between $\mathbb{D}_j^t$ and $\mathbb{D}_k^t$. We have only unlabeled training samples for the target task, making it challenging to precisely predict the classification error for the target task. However, we adopts a easy approach where we determine the sampling probability by examining the difference in empirical distributions between $\mathbb{D}_j^t$ and $\mathbb{D}_k^t$, specifically considering the unlabeled samples. Thus, the parameter ν_{jk} can be learned as follows:

$$\nu_{jk} = \frac{\exp\left(1/\hat{d}_{\mathcal{H}\Delta\mathcal{H}}\left(\mathbb{D}_j^t, \mathbb{D}_k^t\right)\right)}{\Xi}, \tag{19}$$

where Ξ refers to the normalization parameter. By using this normalization term, tasks with less difference in their distributions are more likely to be selected for meta-training [48]. This helps the meta-learning algorithm to emphasize tasks that are more relevant to the task at hand, thus improving its ability to quickly adapt to new tasks that share similar data distributions. When the distribution difference is lesser, it ensures better transfer of information between tasks [49]. Hence, we can create a collection of tasks $\mathbb{S}$ by using a sampling probability and train them by using Algorithm 1.

B. Meta-Training

Meta-training plays a crucial role in training the initialized parameters such that they can quickly adjust to the new task. Meta-training is essentially about training the model to "learn how to learn". It aims to improve the model's ability to generalize and transfer knowledge from one scenario to another, making it more capable of handling unseen scenarios effectively. This process can lead to a more efficient and effective decision-making system for V2N handover in dynamic vehicular environments. The θ can be learned as follows [50]:

$$\theta \leftarrow \arg\min_{\theta} \sum_{(j,k) \in \mathbb{S}} \Upsilon_{jk}(\theta), \tag{20}$$

where

$$\Upsilon_{jk}(\theta) = \hat{E}_j^t(\mathcal{F}_{jk}(\theta)) + \varrho.\hat{d}_{\mathcal{H}\Delta\mathcal{H}}\left(\mathbb{D}_j^t, \mathbb{D}_k^t; \mathcal{F}_{jk}(\theta)\right) \tag{21}$$

refers to the loss function. Furthermore,

$$\mathcal{F}_{jk}(\theta)) \leftarrow \theta - \mho \nabla_\theta L\left(\mathbb{D}_j^t, \mathbb{D}_k^t\right) \tag{22}$$

where $\mathcal{F}_{jk} : \theta \rightarrow \theta_{jk}$ refers to the function which transforms the θ into the optimal parameter for the given task θ_{jk}, and $\mathcal{F}$ is updated by using the gradient descent approach. The adaptation updating capability is evaluated by using (21) on a new task and calculating the loss function according to its corresponding validation set on meta-tasks. Every task incorporates the latest experience by adding the corresponding loss function.

C. Meta-adaptation & testing

Meta-testing refers to the evaluation of a learning model's performance on unseen tasks or scenarios after it has undergone meta-training. When it comes to meta-training, the objective is to train a model to learn how to learn effectively across a range of tasks, enabling it to adapt quickly during meta-testing. The proposed approach can quickly adapt to the latest target task (e.g. new channel conditions) with fewer steps, using training parameters learned from previous source

This article has been accepted for publication in IEEE Transactions on Vehicular Technology. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TVT.2024.3431875

Algorithm 2 A2E Framework

Input: Learning rates, source task $\mathbb{D}_k^s$, batch size and target task $\mathbb{D}_{K+1}^t$.
Output: Meta-trained agent with a HO decision-making policy.

———META-TRAINING———

Initialize meta-tasks $\mathbb{S} = \{\}$, replay buffer $\mathbb{B}$
for each time step **do**
  Create experience set (s_t, a_t, r_t, s_{t+1}) according to Algorithm 1.
  Save experience data into $\mathbb{B}$.
  Sample mini-batch from the $\mathbb{B}$.
  Generate tasks and approximate the ν using (19).
  Update θ according to (20).
  Compute the gradient of the loss function on validation data.
  Create the pseudo-label for the target task.
end for

——META-ADAPTATION AND TESTING——

for each time step **do**
  Create experience sets of data according to Algorithm 1.
  Store the experiences into $\mathbb{B}$.
  Sample mini-batch from the memory and fine-tune on $\mathbb{D}_{K+1}^t$ according to (23).
end for
return Predicted HO time function on the newest target task.

and target tasks. The best way to learn the optimal parameters θ_{K+1} for the latest target task $\mathbb{D}_{K+1}^t$ is by updating the θ on a carefully chosen tasks.

$$\theta_{K+1} = \mathcal{F}_{i(K+1)}(\theta) \leftarrow \theta - \upsilon \nabla_\theta L\left(\mathbb{D}_i^t, \mathbb{D}_{K+1}^t\right), \qquad (23)$$

where θ refers to the initialized parameter learned in the meta-training stage. Once the parameters are adopted for the new target task, we can assess the performance of the proposed algorithm in the testing phase.

D. Algorithm formulation

We introduce a comprehensive framework that addresses the challenges of dynamic HO decision-making in V2N communication. The framework combines transfer learning, meta-learning, and the DDQN-TS algorithm to enable adaptive HO optimization. It leverages historical data, generalizes across time-evolving source and target tasks, and dynamically evolves the HO decision-making strategy to maximize average throughput while minimizing the number of handovers.

First, we collect a set of time-evolving source tasks and a target task and train on the source task using Algorithm 1 to learn a HO time function for the target task. Then, we use the error bound derived in the previous section to estimate the expected generalization error of the HO time function on the target task. A meta-learning objective function is presented

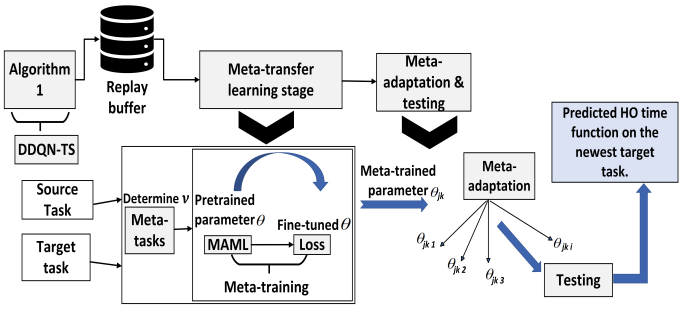

Fig. 3: Block diagram of proposed A2E framework

that minimizes the upper bound of the expected generalization error. We optimize the meta-learning objective function and update the meta-learner parameters θ using stochastic gradient descent. In the adaptation phase, the trained meta-learner is utilized to adapt the HO time function learned on the source task to the current state of the target task. The A2E framework allows the agent to adapt to the target task by leveraging knowledge from the time-evolving source tasks. It enables the centralized DDQN agent to learn effective HO decision-making strategies in dynamic V2N communication environments, maximizing average throughput while minimizing the number of HOs. Algorithm 1 serves as the core DRL algorithm in the process, facilitating efficient and effective learning of the HO policy. Algorithm 2 provide details of the proposed A2E approach. Moreover, Fig. (3) presents the block diagram of the proposed A2E framework.

VI. SIMULATION RESULTS

It is of vital importance to use high-quality datasets to train and evaluate ML algorithms. However, collecting real-world HO data is highly complex and challenging. In this research, we used ns-3 network simulation to generate datasets and evaluate the effectiveness of our proposed solution [51]. The ns-3 simulator is an open-source, discrete-event full-stack simulator that provides realistic simulation and a standardized evaluation platform. It allows the tracing of internal events with flexible configurations and supports multiple communication technologies. To configure an LTE cellular V2N network, we chose ns-3's official standard-compliant LTE module LENA [52]. [2]

For training and testing the algorithm, we configured three scenarios based on the 3GPP specifications for V2X performance evaluations as in Annex A of [55]:

- Urban setting: This scenario's configuration strictly follows the Manhattan grid model for the urban case in Table A-1 of [55]. It contains 9 grids (433 m $\times$ 250 m grid size), 2 lanes (3.5 m in width) in each direction for each grid, and thus a total simulation area size of 1299 m $\times$ 750 m.
- Rural setting: This is highly similar to the urban configuration above but with grid size set to $1000m \times 1000m$ and wider lane width of 5 m.

[2]The reason for this choice was that the 5G and LTE network HO mechanisms are very similar, while the 5G-LENA module [53] (the 5G version of the LENA module), although open for public access, has not implemented the 5G HO mechanism and interfaces, and is currently reusing the LTE X2 interface [54].

This article has been accepted for publication in IEEE Transactions on Vehicular Technology. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TVT.2024.3431875

TABLE I: Simulation configurations

	Urban	Rural	Highway
Properties	Parameters and values		
Simulation time (s)	800	1000	600
LTE Network	5 sites with 3 cells/site		3 sites with 3 cells/site
BS Antenna model	15 dB Cosine model, 65°half power beamwidth		15 dB Cosine model, 40°half power beamwidth
BS height	25 m	35 m	35 m
VUE speed	50 km/h	108 km/h	120 km/h
Pathloss model	3GPP UMa	3GPP RMa	log-distance
BS Transmit power	40 dBm		
Carrier frequency	2115 MHz, downlink only		
Channel bandwidth	2×10 MHz (2×50 RBs)		
Noise figure	BS: 9 dB UE: 5 dB		
Scheduling algorithm	Proportional Fair		
Applicaiton setup	UDP, downlink only packet interval: 20 ms (50 packets/s) individual packet size: 4096 bits		
Data collection frequency	RSRP: every 200 ms Other HO information: Event-triggered		

- Highway setting: Also based on Table A-1 of [55], this scenario is set to 1500 m in length, with 3 lanes in each direction, hence, 6 lanes in total for a highway segment. The lane width is set to 5 m and wrap-around is also implemented.

For VUE mobility, we used the Simulation of Urban Mobility (SUMO) software to generate realistic moving trajectories for the corresponding scenarios. For the urban and rural scenarios, we utilized the Manhattan Grid mobility model as specified in 3GPP TR 37.885 [55], where a VUE goes straight and turns left or right with a probability of 0.5, 0.25, and 0.25, respectively, in an area of street grid. For the Highway scenario, a VUE is randomly placed at a lane, goes straight along that lane, and turns at the end of the road (i.e., wrap-around) to the other direction. To guarantee reproducible results, we configured one trajectory for a VUE to fully explore the simulated area for each scenario, which was manually checked via SUMO's graphical interface during the generation and reflected by varying the simulation times of each scenario. Note that the different simulation times are for the trajectories to cover the simulation areas in each scenario, as recorded in Table I. After generation, the trajectories from SUMO were imported to ns-3 for network simulation.

For the BSs, we randomly set their locations in each scenario with predefined rules to guarantee the overall coverage in each scenario. We also configured each BS to have 3 sectorized antennae with random yet non-overlapping antenna orientations, and we used the cosine antenna model for all BSs. Fig. 4 shows the above configurations in urban scenario including a demonstrative VUE's trajectory, the placement of BSs, and demonstrative antenna orientations with horizontal beam-widths for the red BS.

To also consider small-scale fading, we applied trace-based fading generated via the script provided in the LENA module [56]. The fast fading model is derived from the Jake's Model for Rayleigh fading and this generation approach is considered the official approach for fast fading implementation [57]. The "vehicular" mode was chosen for fading trace generation and the moving speed was set according to the scenarios' setups. As for other network configurations such as the carrier frequency, we set the parameters according to the 3GPP standards

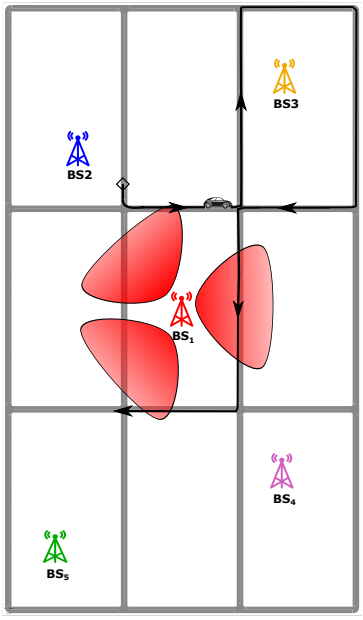

Fig. 4: Illustration of the urban scenario used in the simulation, with 5 BSs placed randomly in the area ensuring coverage and a VUE moving along the arrowed-line trajectory. The background urban grid image is exported from SUMO. For demonstration, the triangles represent the antenna orientations and horizontal beam widths of the sectorized antennae of BS_1.

[55]. Table I lists the detailed network configuration. Note that we also configured different path loss models to reflect different communication environments in each scenario. We introduce the benchmarks where DDQN-TS and DDQN [17] are evaluated based on 1) using similar-scenario for training and testing and 2) different-scenario for training and testing. In [17], the approach utilized for the DDQN involves employing an ϵ-greedy strategy. We evaluate the generalization ability of A2E using different-scenario for training and testing. All learning-based approaches were trained offline with datasets generated via ns-3, and evaluated in an online manner supported by the "ns3-ai" module [58]. At both stages, RSRP data is sampled every 200 ms, while other related information for a HO trigger is recorded when the HO happens, including time stamps and corresponding serving / target cell.

A. Highway setting

Initially, we assess the outcomes of the proposed schemes in a highway scenario. For this, we train the model in the baseline setting (urban environment) and then test it in the highway environment. It is well established that the more the DDQN agent experiences new channel conditions, the better it performs. Yet, gaining enough experience takes a considerable amount of time. Hence, we have proposed the A2E approach to maximize learning with fewer steps, where A2E model trained and tested in different environment. The convergence results presented in the Fig. 5 demonstrate that the DDQN-TS (similar-scenario) algorithm acts as a performance upper bound, achieving the highest normalized reward and demonstrating good convergence behaviour. The A2E algorithm also exhibits the fastest

This article has been accepted for publication in IEEE Transactions on Vehicular Technology. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TVT.2024.3431875

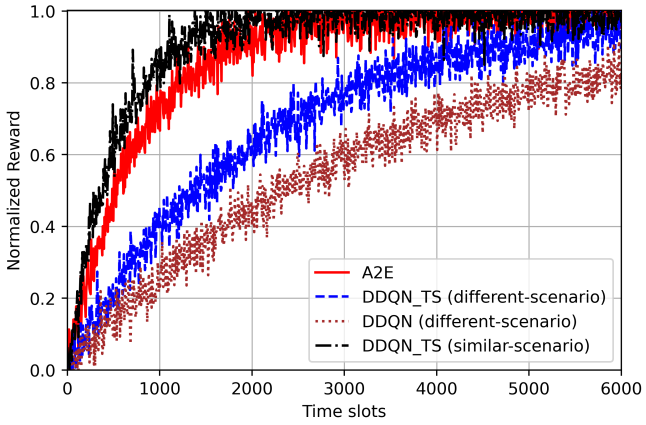

Fig. 5: Convergence analysis

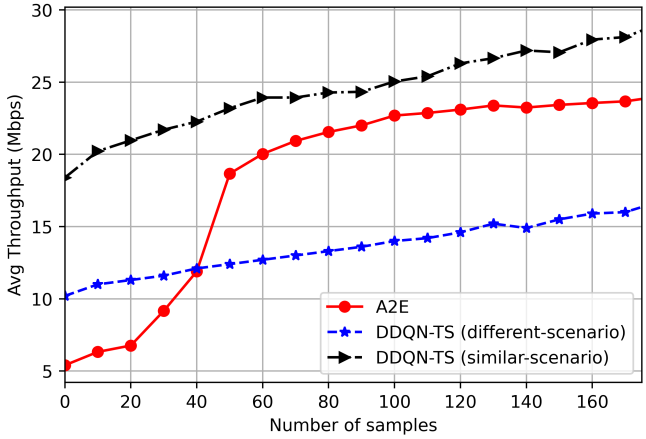

Fig. 6: Learning performance of the proposed methodologies

convergence rate and achieves the highest normalized reward, indicating its superior performance in optimizing handover decisions and quickly adapting to the environment compared to DDQN-TS (different-scenario). Conversely, the DDQN-TS (different-scenario) algorithm shows significant improvements over the standard DDQN, with faster convergence and higher rewards, underscoring the adaptability of TS in varying conditions. The standard DDQN (different-scenario) algorithm, however, exhibits the slowest convergence and lowest rewards, highlighting its limitations in dynamic environments. This result emphasizes the superiority of TS and the efficacy of the A2E algorithm in enhancing the adaptability and performance of handover decision algorithms in rapidly changing vehicular network environments.

Next, we show the analysis of adaptation on the throughput performance of the proposed A2E approach in Fig. 6. We collect samples at each time step. As shown in Fig. 6, the throughput performance gets better with more samples. The A2E approach outperforms the DDQN-TS approach in a different scenario. When there are more than 40 samples, the A2E method performs better in terms of average throughput. A significant difference in performance exists between training a model in similar scenarios and training it in different environments. This gap occurs because the model trained in a different environment serves as the baseline case. The A2E algorithm adapts to new channel conditions after processing 40 samples. Therefore, we will use this sample size for adaptation. Afterwards, we evaluate how Algorithms 1 and 2 perform in the highway setting compared to DDQN (ϵ-greedy) approach. HO triggering moments are presented in Fig. 7, where fading is considered. It can be observed from Fig. 7 that the optimal triggering moment of the HO from BS_2 to BS_3 is around 104 seconds. While striving to minimize repetitive HOs, we sometimes opt to let go of a connection to the BS with the higher SINR, which may trigger a HO shortly.

The proposed A2E approach triggers the HO at around 106.61 seconds, immediately when the optimal triggering moment happened. Whereas, DDQN-TS (different-scenario) triggers the HO at 112.4 seconds, which is trained in the urban setting and tested in the highway setting. On the other

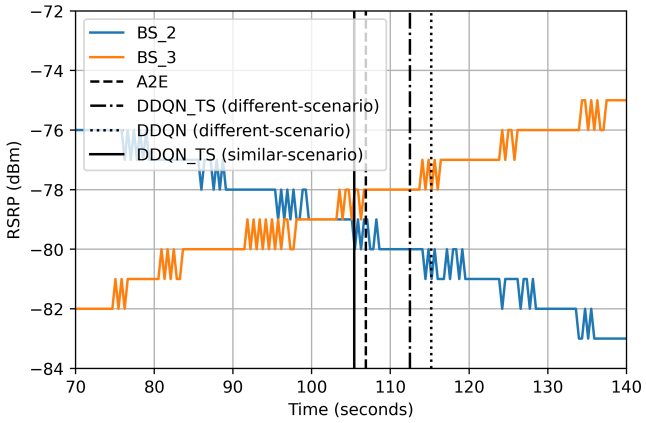

Fig. 7: The triggering moments of proposed algorithms plotted as vertical for a highway HO case

hand, DDQN (ϵ-greedy) triggers the HO at 115.20 seconds. The proposed A2E approach performs better compared to the other approaches. It is because the A2E approach can deal with the effects of environmental changes due to its generalization ability. DDQN-TS performs better than DDQN with the ϵ-greedy approach because the ϵ-greedy strategy might get stuck in local optima and have difficulties in finding the optimal solution. Whereas, TS can adapt to varying uncertainties and perform well due to better exploration properties. DDQN-TS (similar-scenario) provides the performance lower bound as it has been trained and tested in similar-scenario.

Next, we examine the probability of HO which shows the percentage of HOs events. Fig. 8 shows the HO probability for all schemes in the highway scenario. It can be seen that the A2E approach possessing generalization ability considerably reduces the HO probability compared to the other approaches. A2E has achieved HO probability of less than 14%, while the DDQN-TS (different-scenario), and DDQN (different-scenario) obtained HO probabilities of 29%, and 38%, respectively. On the other hand, DDQN-TS (similar-scenario) achieves the HO probability of 11 % providing a performance lower bound because it has been trained and tested in similar communication scenarios. Next, we present

This article has been accepted for publication in IEEE Transactions on Vehicular Technology. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TVT.2024.3431875

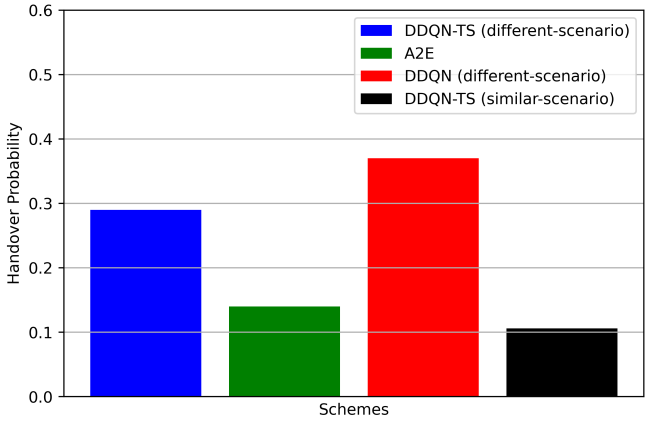

Fig. 8: HO probability in Highway scenario

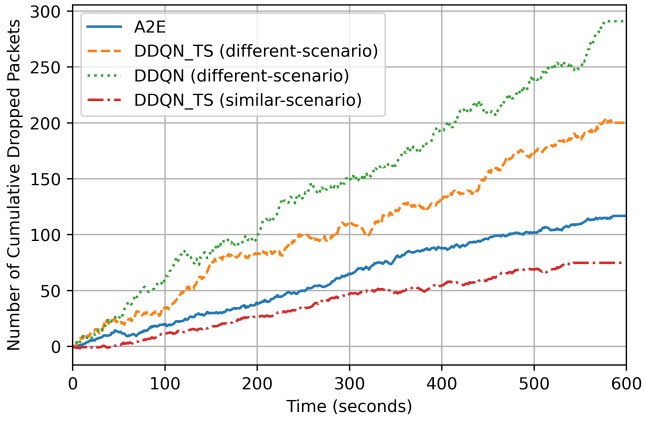

Fig. 10: Packet-loss comparison in the highway scenario

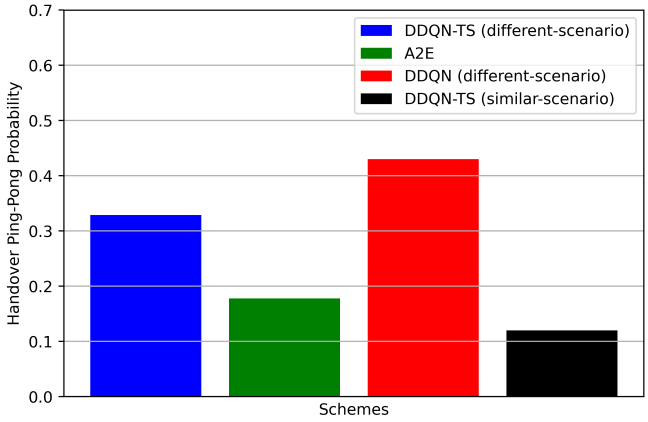

Fig. 9: HO ping-pong probability in the highway scenario

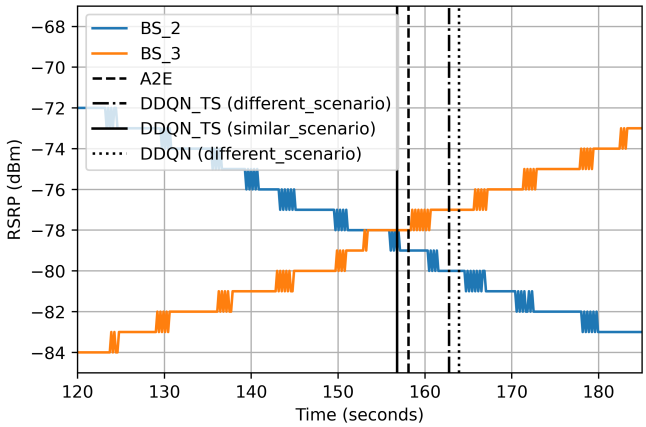

Fig. 11: The triggering moments of proposed algorithms in the rural setting

the HO ping-pong probability in the highway case. The 3GPP [59] defines a metric known as ping-pong rate to assess how effectively a BS manages HOs. In Fig. 9, we show the HO ping-pong probability performance of all the schemes. It can be observed that the proposed A2E approach reduces the ping-pong probability compared to other schemes. A2E achieves a ping-pong probability of 17 %, whereas DDQN-TS (different-scenario) and DDQN (different-scenario) achieve probabilities of 32 % and 43 %, respectively.

The better performance of HO is evident in the reduction of packet loss. In our work, we do not assume the constant availability of unengaged resource blocks at the target base station. We consider the realistic scenario where resource blocks may be fully engaged, leading to potential handover failures and packet drops. The cumulative dropped packets metric provides a comprehensive evaluation, where we have analysed the packet loss resulting from incomplete handovers due to resource block unavailability. This analysis helps to quantify the impact of traffic load on handover performance and the resulting packet loss. In Fig. 10, we show the packet loss comparison where the Packet Data Convergence Protocol (PDCP) packet loss is measured during HOs for the proposed algorithms. The A2E algorithm outperforms other algorithms when considering different scenario for training and testing. It leads to 93 fewer lost packets during all HOs compared to the DDQN-TS (different-scenario), indicating a 48.02 % reduction in cumulated packet loss. Each HO results in an average reduction of 29.32 % in packet loss. Note that our application configuration does not lead to a high packet burden, and more demanding applications can lead to more significant packet loss results.

B. Rural setting

In the previous section, we looked at how well the system performed in the highway communication scenario. Now, we evaluate the suggested methods in the rural setting to see how effective they are. Fig. 11 presents the HO triggering moments in rural settings. It can be seen that the optimal triggering moment of the HO from BS2 to BS3 is around 156 seconds. DDQN-TS (similar-scenario) triggers the HO at 156.8 seconds, providing the performance lower bound where we train and test the algorithm in the same scenario. The proposed A2E and DDQN-TS (different-scenario) trigger the HO at 158.1 and 163.02 seconds, respectively. Whereas, DDQN (different-scenario) performs poorly as it triggers the HO at 164 seconds. The proposed A2E approach provides the gain in time delay of 5 seconds compared to DDQN-TS (different-scenario). It is to be noted that the urban setting has been used to train

This article has been accepted for publication in IEEE Transactions on Vehicular Technology. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TVT.2024.3431875

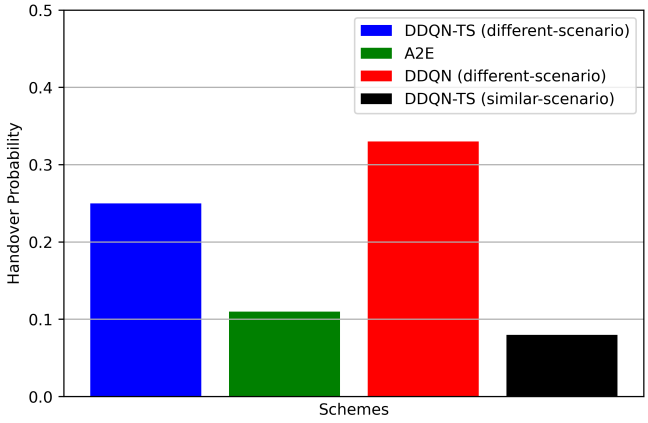

Fig. 12: HO probability in the rural setting

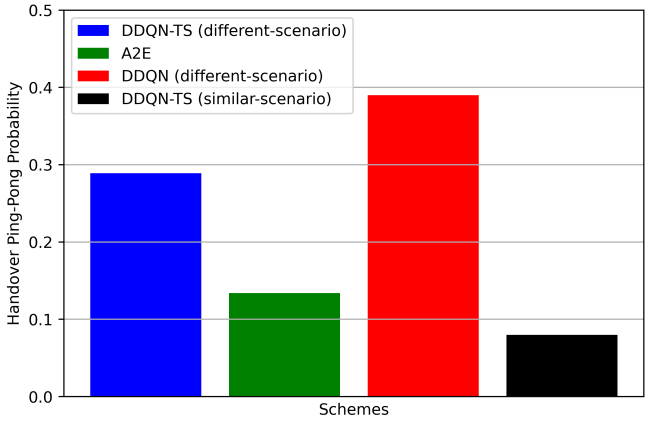

Fig. 13: HO ping-pong probability in the rural setting

the DDQN-TS (different-scenario) approach and tested in the rural setting. Next, we compare the probability of HO based on the algorithms in Fig. 12. The proposed A2E approach considerably reduces the HO probability compared with the DDQN-TS and DDQN under different scenarios by achieving the lowest HO probability of less than 11%. Afterwards, we show the HO ping pong probability in the rural setting in Fig. 13. The proposed A2E approach outperforms the DDQN and DDQN-TS under different scenario by reducing the ping-pong probability to less than 13%. The DDQN and DDQN-TS under different scenario implementation performs poorly due to the lack of generalization ability. Finally in Fig. 14, we present the packet loss comparison in the rural setting during the HO processes. It can be observed that the A2E approach performs better compared to the DDQN and DDQN-TS under different scenario implementations. It leads to 87 fewer lost packets compared to the DDQN-TS (different-scenario), indicating a 46.32 % reduction in cumulated packet loss. This shows that the suggested A2E method can quickly adjust to different channel situations, and has good generalization ability.

VII. Conclusion

In this paper, we have proposed a novel meta-learning framework for optimizing dynamic V2N communication HO using a DRL-based approach. By combining transfer learning,

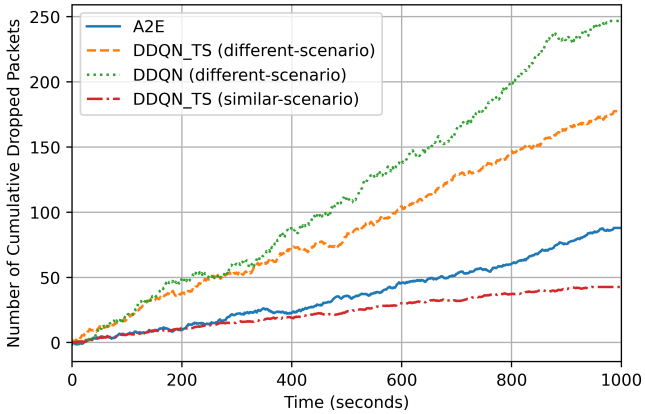

Fig. 14: Packet loss comparison in the rural setting

meta-learning, and the DDQN-TS algorithm, we empower the agent to adapt to dynamic network conditions, optimize HO decisions, and improve overall system performance. The framework addresses the challenges of HO decision-making in V2N scenarios, where the dynamic nature of the environment requires adaptive strategies to maintain seamless connectivity and maximize average throughput while reducing the number of HOs. A key contribution of our work is the derivation of generalization bounds for the expected generalization error of the learned HO time function. These bounds provide valuable insights into the performance and scalability of our approach, giving theoretical guarantees on the effectiveness of the HO decision-making mechanism in varying V2N environments. The proposed A2E algorithm allows the agent to efficiently learn HO decision-making policies. Through extensive experiments using the ns-3 full-stack network simulator with the standard-compliant LENA module and a realistic simulation setup, we have demonstrated the effectiveness of our approach in optimizing dynamic V2N communication HO. With its ability to dynamically evolve and adapt, our framework provides a promising solution to the challenges of HO decision-making in dynamic V2N communication scenarios, contributing to the advancement of intelligent and efficient transportation systems. Future research can pave the way for even more sophisticated and adaptive V2N communication systems by incorporating energy efficiency considerations into the HO optimization process. This may involve introducing energy-related rewards or constraints to encourage the agent to make energy-aware HO decisions.

Appendix

Proof of Theorem 1: According to [60], we define a function f over sample set $\mathcal{D} = \left\{ \{(a_{j0}, b_{j0})\}_{j=1}^{z}, ..., \{(a_{jK}, b_{jK})\}_{j=1}^{z} \right\}$ as:

$$f(\mathcal{D}) = \sup_{h \in \mathcal{H}} E_{K+1}^{t}(h) - \sum_{j=0}^{K} \sum_{K=j+1}^{K+1} \nu_{jk} \hat{E}_{j}^{t}(h) \qquad (24)$$

$$= \sup_{h \in \mathcal{H}} \sum_{j=0}^{K} \sum_{K=j+1}^{K+1} \nu_{jk} \left(E_{K+1}^{t}(h) - \hat{E}_{j}^{t}(h) \right).$$

Assuming $\mathcal{D}$ and $\mathcal{D}'$ are the two sample sets for estimating $\nu_{jk}\hat{E}_j^t(h) = \frac{\nu_{jk}}{z}\sum_{i=1}^{z} L(h(a_{jz}^t), b_{jz}^t)$. Then we have

$$|f(\mathcal{D}) - f(\mathcal{D}')| = \frac{1}{z}\sup_{h\in\mathcal{H}}|L(h(a), b) - L(h(a'), b')| \leq \frac{\nu_{jk}}{z} \tag{25}$$

By using McDiarmid's inequality [61], we analyze how the meta-learner's performance on a new task deviates from its expected performance.

$$\Pr\left[f(\mathcal{D}) - \mathbb{E}_{\mathcal{D}}[f(\mathcal{D})] \geq E\right] \leq \exp\left(\frac{-2zE^2}{\sum_{j=0}^{K}\sum_{K=j+1}^{K+1}\nu_{jk}^2}\right) \tag{26}$$

Next, we apply Hoeffding's inequality which ensures that the model's performance on individual tasks doesn't deviate significantly from the expected performance, given a finite number of training samples.

$$\Pr\left[|E_j^t(h) - \hat{E}_j^t(h)| \geq E\right] \leq 2\exp\left(-2zE^2/Z^2\right) \tag{27}$$

By taking the expected value of the expression for $f(\mathcal{D})$ over the distribution of $\mathcal{D}$ in (24), we have

$$\mathbb{E}_{\mathcal{D}}[f(\mathcal{D})] = \mathbb{E}_{\mathcal{D}}\left[\sup_{h\in\mathcal{H}}\sum_{j=0}^{K}\sum_{K=j+1}^{K+1}\nu_{jk}(E_{K+1}^t(h) - \hat{E}_j^t(h))\right] \tag{28}$$

$$\begin{aligned}
&= \mathbb{E}_{\mathcal{D}}\Bigg[\sup_{h\in\mathcal{H}}\sum_{j=0}^{K}\sum_{K=j+1}^{K+1}\nu_{jk}\Bigg(E_{K+1}^t(h) - E_k^t(h) \\
&+ E_k^t(h) - E_j^t(h) + E_j^t(h) - \hat{E}_j^t(h)\Bigg)\Bigg] \\
&\leq \mathbb{E}_{\mathcal{D}}\left[\sup_{h\in\mathcal{H}}\sum_{j=0}^{K}\sum_{K=j+1}^{K+1}\nu_{jk}(E_k^t(h) - E_j^t(h))\right] \\
&+ \mathbb{E}_{\mathcal{D}}\left[\sup_{h\in\mathcal{H}}\sum_{j=0}^{K}\sum_{K=j+1}^{K+1}\nu_{jk}(E_{K+1}^t(h) - E_k^t(h))\right] \\
&+ \mathbb{E}_{\mathcal{D}}\left[\sup_{h\in\mathcal{H}}\sum_{j=0}^{K}\sum_{K=j+1}^{K+1}\nu_{jk}(E_j^t(h) - \hat{E}_j^t(h))\right] \\
&\leq \sum_{j=0}^{K}\sum_{K=j+1}^{K+1}\nu_{jk}\left(\frac{1}{2}d_{\mathcal{H}}(\mathbb{D}_j^t, \mathbb{D}_k^t) + \xi_{jk}\right) \\
&+ \sum_{j=0}^{K}\sum_{K=j+1}^{K+1}\nu_{jk}\left(\frac{1}{2}d_{\mathcal{H}}(\mathbb{D}_k^t, \mathbb{D}_{K+1}^t) + \xi_{k(K+1)}\right) \\
&+ \sum_{j=0}^{K}\sum_{K=j+1}^{K+1}\nu_{jk}\left(\sqrt{\frac{\log 2/\delta}{2z}}\right) \\
&= \sum_{j=0}^{K}\sum_{K=j+1}^{K+1}\nu_{jk}\Bigg(\frac{1}{2}d_{\mathcal{H}}(\mathbb{D}_j^t, \mathbb{D}_k^t) + \frac{1}{2}d_{\mathcal{H}}(\mathbb{D}_k^t, \mathbb{D}_{K+1}^t) \\
&+ \xi_{jk} + \xi_{k(K+1)}\Bigg) + \sqrt{\frac{\log 2/\delta}{2z}} \\
&\leq \sum_{j=0}^{K}\sum_{K=j+1}^{K+1}\nu_{jk}\Bigg(\frac{1}{2}d_{\mathcal{H}}(\mathbb{D}_j^t, \mathbb{D}_k^t) \\
&+ \frac{1}{2}d_{\mathcal{H}}(\mathbb{D}_k^t, \mathbb{D}_{K+1}^t) + 2\xi\Bigg) + \sqrt{\frac{\log 2/\delta}{2z}} \\
&\leq \sum_{j=0}^{K}\sum_{K=j+1}^{K+1}\nu_{jk}\varrho_{jk}d_{\mathcal{H}}(\mathbb{D}_j^t, \mathbb{D}_k^t) + 2\xi + \sqrt{\frac{\log 2/\delta}{2z}} \\
&\leq \sum_{j=0}^{K}\sum_{K=j+1}^{K+1}\nu_{jk}\varrho_{jk}\hat{d}_{\mathcal{H}}(\mathbb{D}_j^t, \mathbb{D}_k^t) + 4 \\
&\sqrt{\frac{\log(2z) + \log(2/\delta)}{z}} + 2\xi + \sqrt{\frac{\log(2/\delta)}{2z}} \\
&= \sum_{j=0}^{K}\sum_{K=j+1}^{K+1}\nu_{jk}\varrho_{jk}\hat{d}_{\mathcal{H}}(\mathbb{D}_j^t, \mathbb{D}_k^t) + \mathcal{O}\Bigg(\xi \\
&+ \sqrt{\frac{\log(2z) + \log(2/\delta)}{z}}\Bigg)
\end{aligned}$$

where; $\varrho_{jk} = \begin{cases} 1 & \text{if} \quad 1 \leq k \leq K \\ \frac{1}{2}\left(1 + \frac{\sum_{k=0}^{j-1}\nu_{kj}}{\nu_{jk}}\right) & \text{if} \quad k = K+1 \end{cases}$

Therefore,

$$\begin{aligned}
f(\mathcal{D}) &= \sup_{h\in H} E_{K+1}^t(h) - \sum_{j=0}^{K}\sum_{K=j+1}^{K+1}\nu_{jk}\hat{E}_j^t(h) \\
&\leq \mathbb{E}_{\mathcal{D}}[f(\mathcal{D})] + \sqrt{\frac{\sum_{j=0}^{K}\sum_{K=j+1}^{K+1}\nu_{jk}^2\log(1/\delta)}{2z}} \\
&\leq \sum_{j=0}^{K}\sum_{K=j+1}^{K+1}\nu_{jk}\varrho_{jk}\hat{d}_{\mathcal{H}}(\mathbb{D}_j^t, \mathbb{D}_k^t) + \mathcal{O}\Bigg(\xi + \\
&\sqrt{\frac{\log(2z) + \log(2/\delta) + \sum_{j=0}^{K}\sum_{K=j+1}^{K+1}\nu_{jk}^2\log(1/\delta)}{z}}\Bigg)
\end{aligned} \tag{29}$$

Now the hypothesis term. We represent the output of the $\overline{h}(a)$ by 1st order taylor expansion such that

$$h_j(a) - \overline{h}(a) \gtrapprox \nabla_\theta\overline{h}(a)\left(-\wp\frac{1}{z}\sum_{i=1}^{z}\nabla_\theta L(\overline{h}(a_{ij}), b_{ij})\right) \tag{30}$$

Then,

$$\begin{aligned}
\hat{E}_j^t(\overline{h}) &= \hat{E}_j^t(h_j) + \hat{E}_j^t(\overline{h}) - \hat{E}_j^t(\overline{h}_j) \\
&\leq \hat{E}_j^t(\overline{h}_j) + \frac{1}{z}\sum_{i=1}^{z}L(\overline{h}(a_{ij}), b_{ij}) - \frac{1}{z}\sum_{i=1}^{z}L(h_j(a_{ij}), b_{ij}) \\
&\leq \hat{E}_j^t(\overline{h}_j) + \frac{1}{z}\sum_{i=1}^{z}|\overline{h}(a_{ij}) - h_j(a_{ij})| \\
&\leq \hat{E}_j^t(\overline{h}_j) + \frac{1}{z}\sum_{i=1}^{z}\left|\nabla_\theta\overline{h}(a_{ij})\left(\wp\frac{1}{z}\sum_{i=1}^{z}\nabla_\theta L(\overline{h}(a_{ij}), b_{ij})\right)\right| \\
&= \hat{E}_j^t(\overline{h}_j) + \frac{1}{z}\sum_{i=1}^{z}\Bigg||\nabla_\theta\overline{h}(a_{ij})\Bigg(\wp\frac{1}{z}\sum_{i=1}^{z}sign(\overline{h}(a_{ij}) - b_{ij}). \\
&\nabla_\theta\overline{h}(a_{ij})\Bigg)\Bigg|
\end{aligned} \tag{31}$$

$$\leq \hat{E}_j^t(\overline{h}_j) + \frac{\wp}{z^2}\sum_{i=1}^{z}||\nabla_\theta \overline{h}(a_{ij})||.\left|\left|\sum_{i=1}^{z} sign(\overline{h}(a_{ij}) - b_{ij}).\nabla_\theta \overline{h}(a_{ij})\right|\right|$$

$$\leq \hat{E}_j^t(\overline{h}_j) + \frac{\wp}{z^2}\sum_{i=1}^{z}||\nabla_\theta \overline{h}(a_{ij})||.\sum_{i=1}^{z}\left|\left|sign(\overline{h}(a_{ij}) - b_{ij}).\nabla_\theta \overline{h}(a_{ij})\right|\right|$$

By using the Cauchy–Schwarz inequality:

$$\hat{E}_j^t(\overline{h}) \leq \hat{E}_j^t(\overline{h}_j) + \wp\left(\frac{1}{z}\sum_{i=1}^{z}||\overline{h}(a_{ij})||\right)^2$$

It completes the proof.

REFERENCES

[1] L. Figueiredo, I. Jesus, J. T. Machado, J. R. Ferreira, and J. M. De Carvalho, "Towards the development of intelligent transportation systems," in *ITSC 2001. 2001 IEEE intelligent transportation systems. Proceedings (Cat. No. 01TH8585)*. IEEE, 2001, pp. 1206–1211.

[2] J. Misener, "Smart transportation," 2020.

[3] H. Peng, L. Liang, X. Shen, and G. Y. Li, "Vehicular communications: A network layer perspective," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 2, pp. 1064–1078, 2018.

[4] C. Weiß, "V2x communication in europe–from research projects towards standardization and field testing of vehicle communication technology," *Computer Networks*, vol. 55, no. 14, pp. 3103–3119, 2011.

[5] M. Pundir and J. K. Sandhu, "A systematic review of quality of service in wireless sensor networks using machine learning: Recent trend and future vision," *Journal of Network and Computer Applications*, vol. 188, p. 103084, 2021.

[6] D. Flore, "5g v2x the automotive use-case for 5g," 2017.

[7] A. Habibzadeh, S. Shirvani Moghaddam, S. M. Razavizadeh, and M. Shirvanimoghaddam, "Modeling and analysis of traffic-aware spectrum handover schemes in cognitive hetnets," *Transactions on Emerging Telecommunications Technologies*, vol. 28, no. 12, p. e3199, 2017.

[8] M. S. Mollel, A. I. Abubakar, M. Ozturk, S. F. Kaijage, M. Kisangiri, S. Hussain, M. A. Imran, and Q. H. Abbasi, "A survey of machine learning applications to handover management in 5g and beyond," *IEEE Access*, vol. 9, pp. 45 770–45 802, 2021.

[9] J. Carew, "What is reinforcement learning? a comprehensive overview," *A Comprehensive Overview,[online] Available: https://www. techtarget. com/searchenterpriseai/definition/reinforcement-learning*, 2021.

[10] L. Liang, H. Ye, G. Yu, and G. Y. Li, "Deep-learning-based wireless resource allocation with application to vehicular networks," *Proceedings of the IEEE*, vol. 108, no. 2, pp. 341–356, 2019.

[11] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," *arXiv preprint arXiv:1509.02971*, 2015.

[12] N. Aljeri and A. Boukerche, "Mobility management in 5g-enabled vehicular networks: Models, protocols, and classification," *ACM Computing Surveys (CSUR)*, vol. 53, no. 5, pp. 1–35, 2020.

[13] Y. Sun, M. Peng, Y. Zhou, Y. Huang, and S. Mao, "Application of machine learning in wireless networks: Key techniques and open issues," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 4, pp. 3072–3108, 2019.

[14] V. Yajnanarayana, H. Rydén, and L. Hévizi, "5g handover using reinforcement learning," in *2020 IEEE 3rd 5G World Forum (5GWF)*. IEEE, 2020, pp. 349–354.

[15] S. Alraih, R. Nordin, A. A. Samah, I. Shayea, and N. F. Abdullah, "A survey on handover optimization in beyond 5g mobile networks: Challenges and solutions," *IEEE Access*, 2023.

[16] H. Tabassum, M. Salehi, and E. Hossain, "Mobility-aware analysis of 5g and b5g cellular networks: A tutorial," *arXiv preprint arXiv:1805.02719*, 2018.

[17] K. Tan, D. Bremner, J. Le Kernec, Y. Sambo, L. Zhang, and M. A. Imran, "Intelligent handover algorithm for vehicle-to-network communications with double-deep q-learning," *IEEE Transactions on Vehicular Technology*, vol. 71, no. 7, pp. 7848–7862, 2022.

[18] A. Abdelmohsen, M. Abdelwahab, M. Adel, M. S. Darweesh, and H. Mostafa, "Lte handover parameters optimization using q-learning technique," in *2018 IEEE 61st International Midwest Symposium on Circuits and Systems (MWSCAS)*. IEEE, 2018, pp. 194–197.

[19] A. Masri, T. Veijalainen, H. Martikainen, S. Mwanje, J. Ali-Tolppa, and M. Kajó, "Machine-learning-based predictive handover," in *2021 IFIP/IEEE International Symposium on Integrated Network Management (IM)*. IEEE, 2021, pp. 648–652.

[20] T. Goyal and S. Kaushal, "Handover optimization scheme for lte-advance networks based on ahp-topsis and q-learning," *Computer Communications*, vol. 133, pp. 67–76, 2019.

[21] Z. Wang, L. Li, Y. Xu, H. Tian, and S. Cui, "Handover control in wireless systems via asynchronous multiuser deep reinforcement learning," *IEEE Internet of Things Journal*, vol. 5, no. 6, pp. 4296–4307, 2018.

[22] M. Simsek, M. Bennis, and I. Güvenc, "Context-aware mobility management in hetnets: A reinforcement learning approach," in *2015 ieee wireless communications and networking conference (wcnc)*. IEEE, 2015, pp. 1536–1541.

[23] N. Aljeri and A. Boukerche, "A two-tier machine learning-based handover management scheme for intelligent vehicular networks," *Ad Hoc Networks*, vol. 94, p. 101930, 2019.

[24] S. H. Srikantamurthy and A. Baumgartner, "A novel unified handover algorithm for lte-a," in *2021 17th International Conference on Network and Service Management (CNSM)*. IEEE, 2021, pp. 407–411.

[25] M. S. Mollel, A. I. Abubakar, M. Ozturk, S. Kaijage, M. Kisangiri, A. Zoha, M. A. Imran, and Q. H. Abbasi, "Intelligent handover decision scheme using double deep reinforcement learning," *Physical Communication*, vol. 42, p. 101133, 2020.

[26] D. Guo, L. Tang, X. Zhang, and Y.-C. Liang, "Joint optimization of handover control and power allocation based on multi-agent deep reinforcement learning," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 11, pp. 13 124–13 138, 2020.

[27] Y. Sun, W. Jiang, G. Feng, P. V. Klaine, L. Zhang, M. A. Imran, and Y.-C. Liang, "Efficient handover mechanism for radio access network slicing by exploiting distributed learning," *IEEE Transactions on Network and Service Management*, vol. 17, no. 4, pp. 2620–2633, 2020.

[28] K. Qi, T. Liu, and C. Yang, "Federated learning based proactive handover in millimeter-wave vehicular networks," in *2020 15th IEEE International Conference on Signal Processing (ICSP)*, vol. 1. IEEE, 2020, pp. 401–406.

[29] N. AlJeri and A. Boukerche, "An efficient movement-based handover prediction scheme for hierarchical mobile ipv6 in vanets," in *Proceedings of the 15th ACM International Symposium on Performance Evaluation of Wireless Ad Hoc, Sensor, & Ubiquitous Networks*, 2018, pp. 47–54.

[30] Z. Ali, M. Miozzo, L. Giupponi, P. Dini, S. Denic, and S. Vassaki, "Recurrent neural networks for handover management in next-generation self-organized networks," in *2020 IEEE 31st Annual International Symposium on Personal, Indoor and Mobile Radio Communications*. IEEE, 2020, pp. 1–6.

[31] N. Aljeri and A. Boukerche, "A two-tier machine learning-based handover management scheme for intelligent vehicular networks," *Ad Hoc Networks*, vol. 94, p. 101930, 2019.

[32] L. Rabiner and B. Juang, "An introduction to hidden markov models," *ieee assp magazine*, vol. 3, no. 1, pp. 4–16, 1986.

[33] K. Tan, D. Bremner, J. Le Kernec, L. Zhang, and M. Imran, "Machine learning in vehicular networking: An overview," *Digital Communications and Networks*, vol. 8, no. 1, pp. 18–24, 2022.

[34] S. Sesia, I. Toufik, and M. Baker, *LTE-the UMTS long term evolution: from theory to practice*. John Wiley & Sons, 2011.

[35] 3GPP, "Evolved universal terrestrial radio access (e-utra); requirements for support of radio resource management," *3rd Generation Partnership Project (3GPP), TS 36.133*, 2021.

[36] 3rd Generation Partnership Project (3GPP), "Evolved universal terrestrial radio access (e-utra); radio resource control (rrc) protocol specification," *3GPP, TS 36.331*, 2021.

[37] A. Orsino, G. Araniti, A. Molinaro, and A. Iera, "Effective rat selection approach for 5g dense wireless networks," in *2015 IEEE 81st Vehicular Technology Conference (VTC Spring)*, 2015, pp. 1–5.

[38] A. Alhammadi, M. Roslee, M. Y. Alias, I. Shayea, and S. Alraih, "Dynamic handover control parameters for lte-a/5g mobile communications," in *2018 Advances in Wireless and Optical Communications (RTUWO)*. IEEE, 2018, pp. 39–44.

[39] J. Sultan, M. S. Mohsen, N. S. Al-Thobhani, and W. A. Jabbar, "Performance of hard handover in 5g heterogeneous networks," in *2021 1st International Conference on Emerging Smart Technologies and Applications (eSmarTA)*. IEEE, 2021, pp. 1–7.

[40] R. Arshad, H. ElSawy, S. Sorour, T. Y. Al-Naffouri, and M.-S. Alouini, "Handover management in dense cellular networks: A stochastic geometry approach," in *2016 ieee international conference on communications (icc)*. IEEE, 2016, pp. 1–7.

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

[42] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. A. Riedmiller, "Playing atari with deep reinforcement learning," *ArXiv*, vol. abs/1312.5602, 2013.

[43] H. Van Hasselt, A. Guez, and D. Silver, "Deep reinforcement learning with double q-learning," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 30, no. 1, 2016.

[44] E. Rastorgueva-Foi, M. Costa, M. Koivisto, K. Leppänen, and M. Valkama, "User positioning in mmw 5g networks using beam-rsrp measurements and kalman filtering," in *2018 21st International Conference on Information Fusion (FUSION)*. IEEE, 2018, pp. 1–7.

[45] S. Harris and D. Harris, *Digital design and computer architecture*. Morgan Kaufmann, 2015.

[46] O. Chapelle and L. Li, "An empirical evaluation of thompson sampling," *Advances in neural information processing systems*, vol. 24, 2011.

[47] H. Liu, M. Long, J. Wang, and Y. Wang, "Learning to adapt to evolving domains," *Advances in neural information processing systems*, vol. 33, pp. 22 338–22 348, 2020.

[48] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. March, and V. Lempitsky, "Domain-adversarial training of neural networks," *Journal of machine learning research*, vol. 17, no. 59, pp. 1–35, 2016.

[49] Y. Zhang, T. Liu, M. Long, and M. Jordan, "Bridging theory and algorithm for domain adaptation," in *International conference on machine learning*. PMLR, 2019, pp. 7404–7413.

[50] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *International conference on machine learning*. PMLR, 2017, pp. 1126–1135.

[51] G. F. Riley and T. R. Henderson, "The ns-3 network simulator," in *Modeling and tools for network simulation*. Springer, 2010, pp. 15–34.

[52] N. Baldo, M. Miozzo, M. Requena-Esteso, and J. Nin-Guerrero, "An open source product-oriented lte network simulator based on ns-3," in *Proceedings of the 14th ACM international conference on Modeling, analysis and simulation of wireless and mobile systems*, 2011, pp. 293–298.

[53] N. Patriciello, S. Lagen, B. Bojovic, and L. Giupponi, "An e2e simulator for 5g nr networks," *Simulation Modelling Practice and Theory*, vol. 96, p. 101933, 2019.

[54] "Opensim cttc/cerca. nr module, release 2.4."

[55] 3rd Generation Partnership Project, "Study on evaluation methodology of new vehicle-to-everything (v2x) use cases for lte and nr," 3GPP, Tech. Rep. TR 37.885 V15.0.0., 2017.

[56] "ns-3, (accessed 2023), design documentation of the LTE module, fading model."

[57] G. Piro, N. Baldo, and M. Miozzo, "An lte module for the ns-3 network simulator." in *SimuTools*, 2011, pp. 415–422.

[58] H. Yin, P. Liu, K. Liu, L. Cao, L. Zhang, Y. Gao, and X. Hei, "ns3-ai: Fostering artificial intelligence algorithms for networking research," in *Proceedings of the 2020 Workshop on ns-3*, 2020, pp. 57–64.

[59] 3GPP, "3rd generation partnership project," 3GPP, Tech. Rep. TR 36.839 V11.1.0, 2012.

[60] M. Mohri and A. Muñoz Medina, "New analysis and algorithm for learning with drifting distributions," in *Algorithmic Learning Theory: 23rd International Conference, ALT 2012, Lyon, France, October 29-31, 2012. Proceedings 23*. Springer, 2012, pp. 124–138.

[61] C. McDiarmid, "Concentration," in *Probabilistic methods for algorithmic discrete mathematics*. Springer, 1998, pp. 195–248.