Generalized Multi-Objective Reinforcement Learning with Envelope Updates in URLLC-enabled Vehicular Networks

Zijiang Yan, *Graduate Student Member, IEEE* and Hina Tabassum, *Senior Member, IEEE*

***Abstract*—We develop a novel multi-objective reinforcement learning (MORL) framework to jointly optimize wireless network selection and autonomous driving policies in a multi-band vehicular network operating on conventional sub-6GHz spectrum and Terahertz frequencies. The proposed framework is designed to (i) maximize the traffic flow and minimize collisions by controlling the vehicle's motion dynamics (i.e., speed and acceleration), and (ii) enhance the ultra-reliable low-latency communication (URLLC) while minimizing handoffs (HOs). We cast this problem as a multi-objective Markov Decision Process (MOMDP) and develop solutions for both predefined and unknown preferences of the conflicting objectives. Specifically, we develop a novel envelope MORL solution which develops policies that address multiple objectives with unknown preferences to the agent. While this approach reduces reliance on scalar rewards, policy effectiveness varying with different preferences is a challenge. To address this, we apply a generalized version of the Bellman equation and optimize the convex envelope of multi-objective Q values to learn a unified parametric representation capable of generating optimal policies across all possible preference configurations. Following an initial learning phase, our agent can execute optimal policies under any specified preference or infer preferences from minimal data samples. Numerical results validate the efficacy of the envelope-based MORL solution and demonstrate interesting insights related to the inter-dependency of vehicle motion dynamics, HOs, and the communication data rate. The proposed policies enable autonomous vehicles (AVs) to adopt safe driving behaviors with improved connectivity.**

***Index Terms*—Autonomous driving, multi-objective reinforcement learning, multi-band network selection, resource allocation**

I. Introduction

Facilitating ultra-reliable and low-latency vehicle-to-infrastructure (V2I) communications is a fundamental prerequisite for the realization of autonomous and intelligent transportation systems. Different from throughput-oriented conventional communications, ensuring ultra-reliable low latency communications (URLLC) is challenging as it relies on ensuring the signal-to-interference ratio (SINR), data rate, over-the-air/queuing latency, and decoding probability. Conventional radio frequency (RF) alone cannot efficiently meet the stringent URLLC requirement due to its limited coverage and narrow transmission bandwidths. In this context, 6G technology enables combining the conventional sub-6GHz transmissions[1] in conjunction with extremely high frequencies such as THz transmissions, where the former can compensate for the severe path-loss attenuation of THz transmission, and the latter can help overcome the RF spectrum congestion.

On the other hand, the use of Deep Reinforcement Learning (DRL) is becoming critical for online decision making in highly random, mobile-oriented wireless environments. In the context of V2I communications, a wealth of research has focused on improving network quality of service (QoS) (e.g., including transmission delay, link throughput, etc.) via DRL-based resource allocation [2]–[4]. This research has focused on considering subchannel and power allocation to improve V2I communication. In particular, the authors in [2], [3], [5] adopted deep-Q network (DQN) and multi-agent DQN to simultaneously improve the overall throughput of V2I links and the payload delivery rate of vehicle-to-vehicle (V2V) links. Xu *et al.* [4] derived the contribution-based dual-clip proximal policy to optimise V2I and V2V connections separately. However, their system model includes only a single BS where handovers (HOs) are not considered.

Recently, the authors in [6]–[10] formulated similar optimization tasks as multi-objective optimization problem (MOOP) and proposed to use multi-objective reinforcement learning (MORL) solutions. Hu *et al.* [6] implemented Double-Loop Learning (DLL) to minimize the latency of real-time services transmission and maximize the throughput of non-instant services transmission. In [7], [8], the authors adopted the weighted Tchebycheff method and weighted-sum-MORL to maximize the fraction between the data rate and the power consumption, respectively. From [9], Guo *et al.* applied the Multi-Agent Proximal Policy Optimization (MAPPO) algorithm to address the joint handover and power allocation problem. In [10], Khan *et al.* utilized the Asynchronous Advantage Actor-Critic (A3C) algorithm to devise a vehicle-RSU association policy, aiming to enhance the mobile user experience by maximizing sum data rate of multiple AVs while ensuring a minimum level of service rate for all AVs.

Along another note, most of the existing research works in the transportation are focused on collision-avoidance [11], [12], safe driving [5], [13], and efficient fuel consumption [13]–[15]. For instance, in [11], the authors applied a RL framework for faster travel and the reward is proportional to the AV's velocity along with a penalty for vehicle collision.

Z.Yan and H.Tabassum were with the Department of Electrical Engineering and Computer Science, York University, Toronto, ON, M3J 1P3 Canada e-mail: {zjiyan,hinat}@yorku.ca.

This research was supported by a Discovery Grant funded by the Natural Sciences and Engineering Research Council of Canada (NSERC).

A preliminary version of this work has been presented at the IEEE Global Communications Conference (GLOBECOM), 2022 [1]

[1]We use sub-6GHz and RF communication interchangeably in this paper.

The action space included acceleration, deceleration, lane changes (LC), and maintain speed, whereas the state space was based on AVs' locations and their respective velocities. In [13], the authors applied DDQN to enhance AVs' driving safety and fuel consumption. The state space included AVs' locations, fuel consumption, and velocities, whereas the actions included speeds and LC of AVs. In [14], the authors applied the Intelligent Driver Model (IDM) and Minimizing Overall Braking Induced by Lane Change (MOBIL) to control steering and lane change. The proposed reward design encourages long traffic, high speed and discourages unnecessary LC. Authors in [15] introduced multi-objective actor-critic to improve the tradeoff between energy consumption and travel efficiency of AVs. The authors derived MO actor-critic to optimize two objectives. Recently, a few optimization-based research works have started looking into this problem where the AV velocity has been optimized using simplified car-following models [16]–[20], but these models are computationally expensive, do not learn from past experiences thus not relevant for faster decision-making, and are not applicable to dynamic traffic scenarios with various control and communication parameters. To date, none of the existing research works [2]–[4], [6]–[11], [13]–[18], [23] have considered the inter-dependency of the AV motion dynamics to wireless data rates.

Different from the existing research, our contributions can be explained from two perspectives, i.e., **(1)** jointly optimizing both autonomous driving policies and telecommunication rewards *with and without predefined preferences*, and **(2)** proposing a novel multi-objective *MO-DDQN-Envelope* reinforcement learning solution that enables optimizing policies across varying preferences (i.e., without predefined preferences) in the multi-objective domain. Table I summarizes the distinction of existing research in terms of computational complexity, methodology, and performance metrics. Our contributions can be summarized as follows:

- We develop an MORL framework to design joint network selection and autonomous driving policies in a multi-band vehicular network (VNet). The objectives are to **(i)** maximize the traffic flow and minimize collisions by controlling the vehicle's motion dynamics (i.e., speed and acceleration) from a transportation perspective, and **(ii)** maximize the data rates and minimize handoffs (HOs) by jointly controlling the vehicle's motion dynamics and network selection from telecommunication perspective. We consider a novel reward function that maximizes data rate and traffic flow, ensures traffic load balancing across the network, penalizes HOs, and unsafe driving behaviors.
- The considered problem is formulated as a multi-objective Markov decision process (MOMDP) that has two-dimensional action space and rewards consist of telecommunication and autonomous driving utilities. We jointly optimizing both autonomous driving policies and telecommunication rewards *with and without predefined preferences* using multi-objective reinforcement learning (MORL) approaches.
- We propose a novel multi-objective *MO-DDQN-Envelope* reinforcement learning solution that enables optimizing policies across varying preferences (i.e., without predefined preferences) in the multi-objective domain. Our framework simultaneously optimizes telecommunication and transportation objectives in dynamic environments, balancing trade-offs like collision avoidance, velocity management, handover optimization, and network availability. Unlike scalarized methods, our **MO-DDQN-Envelope** approach dynamically adjusts preferences, mitigating biases and errors introduced by fixed weightings.
- We develop a novel simulation testbed that emulates multi-band wireless network-enabled VNet *RF-THz-Highway-Env* based on *highway-env* [24]. This test environment not only inherits the advantages of autonomous driving, and lane changes on the highway from [24], but also implements RF/THz channel propagation modeling, network selection, and HO control.
- Numerical results shows that the proposed solution outperforms weighted sum-based MORL solutions with DQN by 12.7%, 18.9%, and 12.3% on average transportation reward, average communication reward, and average HO rate, respectively.

The rest of this work is organized as follows. Section II shows the system model, and Section III provides MOMDP formulation. Section IV introduces the proposed solution. The simulations are presented in Section V, and Section VI concludes this research work.

II. System Model and Assumptions

A multi-band downlink network comprising n_R Radio Frequency Base Stations (RBSs) and n_T Terahertz Frequency Base Stations (TBSs) is considered. A multi-vehicle network is also considered, with a multi-lane road comprising N_L lanes. M AVs receive information from the BSs (deployed alongside the road) through V2I communications (as depicted in Figure 1). Each AV is permitted to associate with only one BS at a time, regardless of whether the BS is an RBS or TBS. The on-board units (OBUs) on the AVs receive real-time information from the VNet, including the velocity, acceleration, and lane position of surrounding vehicles. Each RBS and TBS has a bandwidth available to it, represented by W_R and W_T, respectively. Each RBS and TBS is capable of supporting a maximum number of AVs, denoted by Q_R and Q_T, respectively. All AVs are equipped with a single antenna.

Compared to the Gipps' model [25] and learning-based approaches [26], we adopt a transportation model that combines **Kinematics + IDM + MOBIL**, which is widely used in autonomous driving decision-making, as in [14]. This hybrid model provides improved computational efficiency and adaptability to MORL, outperforming the Gipps and learning-based models in these aspects. For the telecommunication model, in contrast to RF-only 5G V2I, THz-only 6G V2I [1], and RIS-assisted models [27], we adopt a hybrid multiband RF–THz V2I framework [28]. This model offers superior performance in terms of throughput, latency, reliability, mobility management, and HOs control.

Ref.	Autonomous Driving				Vehicular Communication				ML Optimization Method	Computational Complexity
	Collision Avoidance	Speed Management	Driving Behaviour	Lane Changes	Mobility Aware URLLC	Handover Management	Interference Aware	Network Availability		
[1]	✓	✓	✓			✓	✓	RF + THz Multibands	Q + DQN	Low
[13]	✓	✓		✓				AGWN	DQN + DDQN	Low
[2]							✓	Single Cell system 2 GHz	DQN	Low
[4]					✓		✓	Multi-Platoon Vehicular	MORL: CD-PPO	High
[15]	✓	✓		✓				RF	MORL: MOAC	High
[21]						✓	✓	UAV Assisted MEC	EMORL	High
[22]	✓	✓		✓				AGWN	MORL: thresholded lexicographic	High
This paper	✓	✓	✓	✓	✓	✓	✓	RF + THz Multibands	MORL MO-DDQN-Envelope	Low

Table I: Comparison between Related Works and This Work

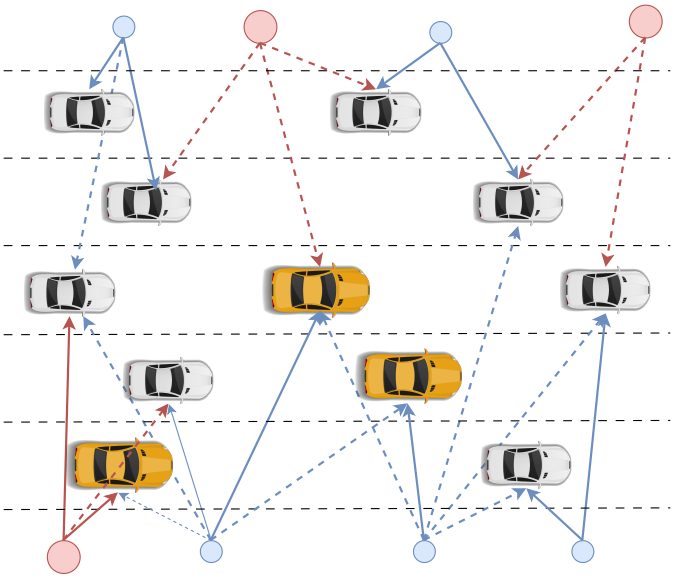


Figure 1: The diagram illustrates the structure of the multi-band VNet model. The blue and red circles represent TBSs and RBSs, respectively. The solid and dashed lines represent desired signal links and interference links, respectively.

A. Downlink V2I Data Transmission Model

The signal transmitted by the RBS is subject to path-loss and short-term channel fading. Subsequently, the signal-to-interference-plus-noise ratio (SINR) of the j-th AV from i-th RBS is given as [28], [29]:

$$\mathrm{SINR}_{ij}^{\mathrm{RF}} = \frac{P_R^{\mathrm{tx}}\, G_R^{\mathrm{tx}}\, G_R^{\mathrm{rx}} \left(\frac{c}{4\pi f_R}\right)^2 H_i}{r_{ij}^{\alpha}\left(\sigma^2 + I_{R_j}\right)}, \tag{1}$$

where P_R^{tx}, G_R^{tx}, G_R^{rx}, c, f_R, r_{ij}, and α represent the transmit power of the RBSs, the gain of the transmitting antenna, the gain of the receiving antenna, the speed of light, the RF carrier frequency in GHz, the distance between the j-th AV and the i-th RBS, and the path-loss exponent, respectively. Note that $r_{ij} = \left(d_{ij}^2 + h_{ij}^2\right)^{1/2}$, where d_{ij} is the 2D distance between the j-th AV and i-th BS and h_{ij} is the transmit antenna height. Furthermore, H_i denotes the exponentially distributed channel fading power observed by the j-th AV from the i-th RBS, σ^2 is the power of the thermal noise at the receiver, and I_{R_j} is the cumulative interference experienced by the j-th AV from other interfering RBSs. $I_{R_j} = \sum_{k\neq i} P_R^{\mathrm{tx}} \gamma_R r_{kj}^{-\alpha} H_k$ where r_{kj} is the distance between the k-th interfering RBS and the j-th AV, H_k is the power of fading from the k-th interfering RBS to the j-th AV, and $\gamma_R = G_R^{\mathrm{tx}}\, G_R^{\mathrm{rx}} \left(c/4\pi f_R\right)^2$.

In the context of a Terahertz (THz) network, where molecular absorption significantly impacts signal propagation, the significance of line-of-sight (LOS) transmissions over non-line-of-sight (NLOS) transmissions is dominant. Consequently, the SINR for a given j-th AV can be modeled as follows:

$$\mathrm{SINR}_{ij}^{\mathrm{THz}} = \frac{G_T^{\mathrm{tx}} G_T^{\mathrm{rx}} \left(\frac{c}{4\pi f_T}\right)^2 P_T^{\mathrm{tx}} \exp(-K_a(f_T) r_{ij}) r_{ij}^{-2}}{N_{T_j} + I_{T_j}}, \tag{2}$$

where $G_T^{\mathrm{tx}}, G_T^{\mathrm{rx}}, P_T^{\mathrm{tx}}, f_T, r_{ij},$ and $K_a(f_T)$ represent the transmit antenna gain of the TBS, the receiving antenna gain of the TBS, the transmit power of the TBS, the THz carrier frequency, the distance between the j-th AV and the i-th TBS, and the molecular absorption coefficient of the transmission medium, respectively [2]. It is important to note that: $G_T^{\mathrm{rx}}(\theta)$ and $G_T^{\mathrm{tx}}(\theta)$ denote the antenna gains at the receiver and transmitter sides corresponding to the boresight direction angle $\theta \in [-\pi, \pi)$. The beamforming gain from the main and side lobes of the TBS transmitting antenna is subsequently defined as,

$$G_T^q(\theta) = \begin{cases} G_{\max}^q & |\,\theta\,| \leq w_q \\ G_{\min}^q & |\,\theta\,| > w_q \end{cases}, \tag{3}$$

where the superscript q is used to indicate the transmit/ receive antenna, i.e., $q \in \{\mathrm{tx}, \mathrm{rx}\}$, w_q is the beamwidth of the main lobe, $G_{\max}^q$ and $G_{\min}^q$ are the beamforming gains of the main and side lobes, respectively. We assume that AVs can align the receiving beam with the TBS transmit beam using beam alignment techniques. For the alignment between the user and interfering TBSs, we define a random variable Θ, $\Theta \in \{G_{\max}^{\mathrm{tx}} G_{\max}^{\mathrm{rx}}, G_{\max}^{\mathrm{tx}} G_{\min}^{\mathrm{rx}}, G_{\min}^{\mathrm{tx}} G_{\max}^{\mathrm{rx}}, G_{\min}^{\mathrm{tx}} G_{\min}^{\mathrm{rx}}\}$, and the respective probability for each case is $F_{\mathrm{tx}} F_{\mathrm{rx}}$, $F_{\mathrm{tx}}(1-F_{\mathrm{rx}})$, $(1-F_{\mathrm{tx}})F_{\mathrm{rx}}$, and $(1-F_{\mathrm{tx}})(1-F_{\mathrm{rx}})$, where $F_{\mathrm{tx}} = \frac{\theta_{\mathrm{tx}}}{2\pi}$ $F_{\mathrm{rx}} = \frac{\theta_{\mathrm{rx}}}{2\pi}$ with $\theta_{\mathrm{tx}}, \theta_{\mathrm{rx}}$ being the beamwidth on the transmitter and receiver antenna, respectively. Without loss of generality, we consider negligible side lobe gains. Thus, the

[2]For the sake of brevity, the argument of $K_a(f_T)$ will henceforth be omitted in this study.

cumulative interference I_T between AV and the interfering TBS is given as $I_T = \sum_{k\neq i} \gamma_T \, P_T^{\mathrm{tx}} \, F_{\mathrm{tx}} F_{\mathrm{rx}} r_{kj}^{-2} \exp(-K_a \, r_{kj})$, where $\gamma_T = G_T^{\mathrm{tx}} G_T^{\mathrm{rx}} \left(\frac{c}{4\pi f_T}\right)^2$. The cumulative thermal and molecular absorption noise is thus given as:

$$N_{T_j} = N_0 + \; P_T^{\mathrm{tx}} \gamma_T \, r_{ij}^{-2} \, (1 - e^{-K_a \, r_{ij}}) + \sum_{k\neq i} \gamma_T F_{\mathrm{tx}} F_{\mathrm{rx}} \, P_T^{\mathrm{tx}} \, r_{kj}^{-2} (1 - e^{-K_a \, r_{kj}}). \tag{4}$$

The traditional data rate relies on Shannon's capacity, which can be attained as the block-length of channel codes approaches infinity. Nevertheless, to prevent prolonged transmission delays in URLLC, the block length must be limited. Consequently, Shannon's capacity cannot be realized due to the presence of a non-zero decoding error probability [30]. From [31], the achievable rate in the short block-length regime over an AWGN channel can be approximated as:

$$R_{ij} = \frac{W_j}{\ln 2} \left[\ln(1 + \mathrm{SINR}_{ij}) - \sqrt{\frac{V}{L_B}} f_Q^{-1}(\epsilon_c) \right] \tag{5}$$

where $W_j, L_B, \epsilon_c, f_Q^{-1}(\cdot), V$ are the transmission bandwidth of BS j, blocklength, decoding error probability, the inverse Q function, and the channel dispersion, respectively. V can be calculated by $1 - \frac{1}{(1+\mathrm{SINR}_{ij})^2}$. Given that D_t time to transmit L_B symbols, the time and frequency resources can be computed by $D_t W = L_B$. where $W = W_R$ for RBSs and $W = W_T$ for TBSs. As the block length L_B approaches infinity, the achieve rate in (5) reaches Shannon's capacity.

Each AV maintains a list of the BSs in terms of the achievable data rate R_{ij} and then informs these BSs. Consequently, each BS can calculate the possible AV associations at each time instance denoted by n_i. Then, the AV collects the traffic load information from these BSs (i.e., the number of AVs associated with each BS n_i). Based on the quota of each BS i, $Q_i \in [Q_R, Q_T]$, each AV computes a *weighted data rate metric* that encourages traffic load balancing at each BS and discourages unnecessary HOs, i.e.,

$$\mathrm{WR}_{ij} = \frac{R_{ij}}{\min(Q_i, n_i)} (1 - \mu) \tag{6}$$

where μ denotes the HO penalty to discourage unnecessary HOs that is defined as follows:

$$\mu = \begin{cases} 0.1, & \text{if switch to a RBS} \\ 0.5, & \text{if switch to a TBS} \\ 0, & \text{keep previous BS} \end{cases} \tag{7}$$

As AVs traverse the corridor, they transition from one BS to another, which is referred to as a HOs. We distinguish between two types of HOs: horizontal and vertical. Horizontal HO denotes the AV connection shifting from one BS of the same type to another. In contrast, vertical HO pertains to the scenario where the AV connection transitions from one specific type of BS to a distinct type of BS, such as moving from an RBS to a TBS. It is evident that frequent HOs can have a significant impact on the data rate that AVs receive, due to the inherent latency and failure rates associated with HOs. In this paper, we propose the introduction of a penalty, denoted by the parameter μ, which is designed to discourage HOs. This penalty is higher for TBSs and lower for RBSs, reflecting the fact that THz transmission is limited to a relatively short distance, rendering it more vulnerable to unnecessary HOs.

Then, each AV prepares a sorted list of BSs offering the best weighted data rates WR_{ij} and associates to those that can fulfill the data rate requirement of the AV given by R_{th}.

B. Transportation Model

We categorize M AVs into two groups: target vehicles, denoted as M_1, and surrounding vehicles, denoted as M_2.

Following [24], [32], we update the real-time physical location of all AVs using the *Kinematics* model [32]. Assuming only the front wheels can be steered, the dynamics of each AV j, $j \in M$, are described as:

$$\frac{\partial x_j}{\partial t} = v_j \cos(\psi_j + \beta_j), \quad \frac{\partial y_j}{\partial t} = v_j \sin(\psi_j + \beta_j), \quad \frac{\partial v_j}{\partial t} = a_j, \tag{8}$$

where (x_j, y_j) represents the position of AV j, v_j its velocity, a_j its commanded acceleration, ψ_j its heading angle, and β_j the slip angle at the vehicle's center of gravity. The slip angle β_j is given by:

$$\beta_j = \arctan\left(\frac{\tan \delta_j^{\mathrm{fa}}}{2}\right), \tag{9}$$

where δ_j^{fa} is the steering angle of the front wheels.

The heading dynamics of AV j depend on its control model:

$$\frac{\partial \psi_j}{\partial t} = \begin{cases} K_j^{\psi} \left[\psi_{L_j} + \arcsin\left(\frac{\tilde{v}_{j,y}}{v_j}\right) - \psi_j \right], & \text{if } j \in M_1, \\ \frac{v_j}{l_j} \sin \beta_j, & \text{if } j \in M_2, \end{cases} \tag{10}$$

where l_j is the half-length of AV j, ψ_{L_j} is the desired heading, and K_j^{ψ} and K_j^{y} are control gains. The term $\tilde{v}_{j,y}$, representing lateral velocity adjustment, is given as:

$$\tilde{v}_{j,y} = K_j^{y} (y_{L_j} - y_j),$$

with y_{L_j} being the lateral position of the desired lane.

The acceleration a_j for AV j is defined as:

$$a_j = \begin{cases} K_0^{v}(v_r - v_j), & \text{if } j \in M_1, \\ a_c - a_c \left[\left(\frac{|v_j|}{v_0}\right)^{\delta_a} + \left(\frac{\hat{d}_j}{d_j}\right)^2 \right], & \text{if } j \in M_2, \end{cases} \tag{11}$$

where v_r is the desired speed, K_0^{v} is the speed control gain, a_c is the maximum acceleration, v_0 is the desired velocity, δ_a is the acceleration reduction factor, and d_j is the distance to the front AV. The desired gap $\hat{d}_j$ is computed as:

$$\hat{d}_j = d_0 + \max\left(0, T v_j + \frac{v_j \Delta v_j}{2\sqrt{a_c b_c}}\right), \tag{12}$$

where d_0 is the minimum distance in stopped traffic, T is the safe time gap, Δv_j is the relative velocity to the front vehicle, and b_c is the comfortable braking deceleration.

For AVs in M_2, discrete lane changes are determined by the MOBIL model [33], [34], where an AV decides to change lanes when (1) AVs are safe to cut-in another lane as $\tilde{a}_j \geq$

$-b_{\text{safe}}$. (2) there is an incentive benefit if the target AVs and followers satisfies $\tilde{a}_j - a_j + p\,(\tilde{a}_o - a_o + \tilde{a}_n - a_n) \geq \Delta a_{\text{th}}$. where $-b_{\text{safe}}$ is the maximum braking imposed on the new following AV, p is the politeness coefficient, Δa_{th} is the minimum acceleration gain, and subscripts o and n denote the old and new followers, respectively.

III. MOMDP Formulation

This section formulates the multi-objective Markov Decision Process (MOMDPs) for the considered problem. We then discuss the design of state-action space and rewards function. The state transitions and rewards are a function of the AV environment and actions taken by the AV.

A. MOMDPs and Pareto Front

The goal of MORL is to obtain policies among M conflicting or competing objectives, where the relative importance (preferences) of each objective may be known or unknown to the agent. Similar to RL, MORL can be formulated by MOMDP which extends the MDP by defining a new reward space, preference space, and preference function, i.e., $\mathcal{M} =< \mathcal{S}, \mathcal{A}, \mathbf{r}, \mathcal{P}, \Omega, \iota_0 >$, where $\mathbf{r} \in \mathbb{R}^H$ is a vector of reward functions corresponding to H objectives, e.g., $\mathbf{r} = [r^1, r^2, \ldots, r^H]$, Ω is the preference space where $\boldsymbol{\omega} \in \Omega$ is the preference vector corresponding to H objectives, and $\sum_{h=1}^{H} \omega^h = 1$. ι_0 is the probability distribution over initial states. In MOMDPs, a policy $\pi : \mathcal{S} \to \mathcal{A}$ defines a mapping from states to actions with the goal of maximizing a vector of expected rewards. $\mathcal{P}(s_{t+1}|s_t, a_t)$ indicates the transition probability for the agent to take an action $a_t \in \mathcal{A}$ on state $s_t \in \mathcal{S}$ to the next state $s_{t+1} \in \mathcal{S}$ in time step t. Given the distribution ι_0 and a policy π, the expected discounted return is defined as:

$$\mathbf{Q}_\pi(s, a, \boldsymbol{\omega}) = \mathbb{E}_\pi \left[\mathbf{r}(s_t, a_t) + \gamma \mathbf{Q}_\pi(s_{t-1}, a_{t-1}, \boldsymbol{\omega})\right] \quad (13)$$

where $\mathbf{r}(s_t, a_t)$ is the immediate vector valued reward at time step t for H objectives. Maximizing the expected reward involves solving the following MOO problem $\max \mathbf{Q}_\pi = \max_\pi [Q_\pi^1, Q_\pi^2, \ldots, Q_\pi^H]$.

A policy π strictly dominates another policy π' if π achieves values at least as high as π' in all objectives and strictly higher in at least one objective:

$$\pi > \pi' \iff \forall h, V_\pi^h \geq V_{\pi'}^h \wedge \exists\, h, V_\pi^h > V_{\pi'}^h \quad (14)$$

Furthermore, a policy π weakly dominates another policy π', if π achieves values greater than or equal to π' in all objectives, i.e., $\pi \geq \pi'$, when $V_\pi^h \geq V_{\pi'}^h, \quad \forall h$. A policy π is considered Pareto-optimal (or non-dominated) if it is not strictly dominated by any other policies.

Considering all returns from MOMDP, we have Pareto frontier set $\mathcal{F}^* := \{\hat{\mathbf{r}} \mid \nexists\, \hat{\mathbf{r}}' \geq \hat{\mathbf{r}}\}$ [35], where $\hat{\mathbf{r}} = \sum_{t=0}^{\infty} \gamma \cdot \mathbf{r}(s_t, a_t)$. For all possible preferences in Ω, we define a convex coverage set (CCS) of the Pareto frontier which contains all returns that provide the maximum cumulative reward, i.e.,

$$\text{CCS} := \left\{\hat{\mathbf{r}} \in \mathcal{F}^* \mid \exists \boldsymbol{\omega} \in \Omega, \forall \hat{\mathbf{r}}' \in \mathcal{F}^* \text{ s.t. } \boldsymbol{\omega}^T \hat{\mathbf{r}} \geq \boldsymbol{\omega}^T \hat{\mathbf{r}}'\right\} \quad (15)$$

where $(\cdot)^T$ denotes the transpose operator.

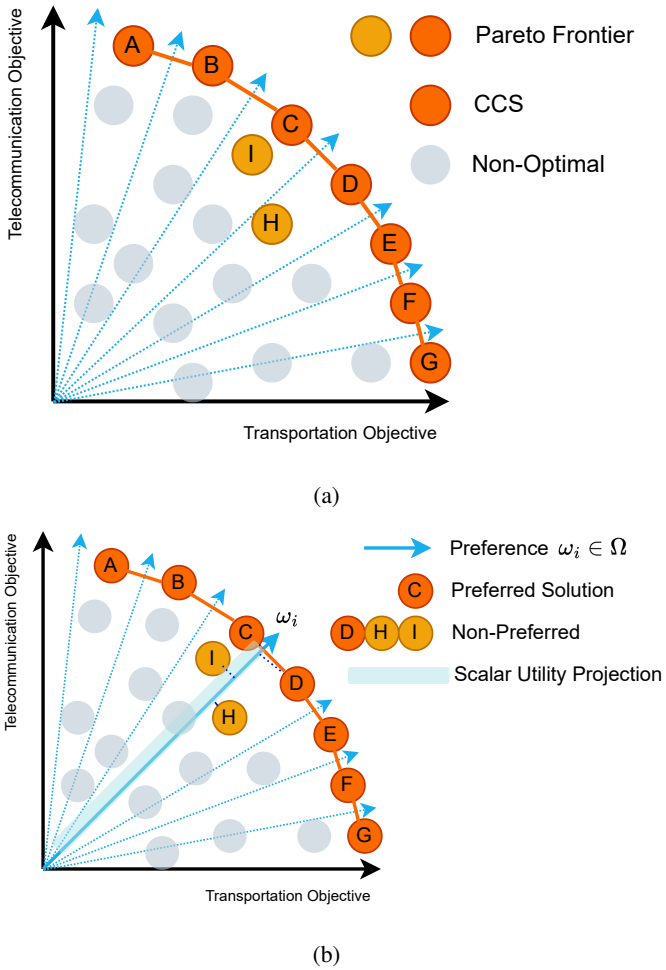


Figure 2: Pareto Frontier and CCS Analysis

From Fig. 2(a), the Pareto frontier includes points A to H, encapsulating some local concave regions. Grey points indicate non-optimal solutions that do not belong to the CCS in terms of transportation and telecommunication objective. In Fig. 2(b), linear preferences are used to select the optimal solution from the CCS based on the highest utility. This utility corresponds to the projection length of each point onto the preference vector $\omega_i \in \Omega$. Blue arrows represent different linear preferences reflecting trade-offs between transportation and telecommunication objectives. Among the possible returns, point C achieves a higher cumulative utility compared to points D, I, and H when projected along the preference vector represented by the solid line ω_i. $\omega_i \cdot \hat{r}_C > \omega_i \cdot \hat{r}_D > \omega_i \cdot \hat{r}_I > \omega_i \cdot \hat{r}_H$. the optimal solution is the point in the CCS with the largest projection along the preference vector ω_i

B. State Space

The state space consists of kinematics-related features, which is a $M_1 \times F$ array that describes $F \to \{x_j, y_j, v_j, \psi_j, n_R^j, n_T^j\}$ specific features of AVs. We consider M_1 target AVs and M_2 surrounding AVs. Each target AV is characterized by its (1) coordinates (x_j, y_j), (2) forward velocity v_j, (3) heading ψ_j, and (4) n_R^j and n_T^j which are the

number of RBSs and TBSs that makes AV achieves the desired data rate in a predefined radius from its current position, respectively. Accordingly, the aggregated state space $\mathcal{S}$ at any time step t is given by:

$$\mathcal{S} = \begin{bmatrix} x_1 & y_1 & v_1 & \psi_1 & n_R^1 & n_T^1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_{M_1} & y_{M_1} & v_{M_1} & \psi_{M_1} & n_R^{M_1} & n_T^{M_1} \end{bmatrix}$$

C. Two Dimensional Action Space

The action space consists of self-driving action space $\mathcal{A}_{\text{tran}}$ and telecommunication action space $\mathcal{A}_{\text{tele}}$, which include 5 and 3 discrete actions, respectively. For each time step t, the AV must select driving-related action and telecommunication-related action from action space, as shown below:

$$\mathcal{A} = \begin{bmatrix} \{a_{\text{tele}}^1, a_{\text{tran}}^1\} & \{a_{\text{tele}}^1, a_{\text{tran}}^2\} & \dots & \{a_{\text{tele}}^1, a_{\text{tran}}^5\} \\ \vdots & \vdots & \vdots & \vdots \\ \{a_{\text{tele}}^3, a_{\text{tran}}^1\} & \{a_{\text{tele}}^3, a_{\text{tran}}^2\} & \dots & \{a_{\text{tele}}^3, a_{\text{tran}}^5\} \end{bmatrix}$$

Note that $\mathcal{A}_{\text{tran}} = \{a_{\text{tran}}^1, \dots, a_{\text{tran}}^5\}$, where $a_{\text{tran}}^1, a_{\text{tran}}^2$ and a_{tran}^3 indicate that AV changes its lane to the left, maintains the same lane, and changes its lane to the right, respectively. a_{tran}^4 and a_{tran}^5 indicate the acceleration and deceleration of AV within the same lane. It is important to note that the acceleration and deceleration rates are dynamically determined by the model in Section II-B. With that being said, each AV selects the same actions does not imply that they will perform identical accelerations/deceleration. The communication action space is represented as $\mathcal{A}_{\text{tele}} = \{a_{\text{tele}}^1, a_{\text{tele}}^2, a_{\text{tele}}^3\}$. a_{tele}^1 indicates scenarios where AV selects a BS by maximizing *weighted data rate metric* (defined by equation (6) in Section II-A), which encourages traffic load balancing between BSs and discourages unnecessary HOs, especially for TBSs. In a_{tele}^2, the AV selects a BS with maximum WR_{ij} by substituting $\mu = 0$, if $Q_i \geq n_i$. Otherwise, AV recursively selects the next vacant best-performing BS in terms of WR_{ij}. In a_{tele}^3, the AV chooses to connect to a BS with the maximum data rate R_{ij}.

D. Rewards

The design of the associated reward is directly related to optimizing the driving policy and network selection, and is critical for accelerating the convergence of the model. Generally, the AV is given a positive reward when it receives a higher HO-aware data rate while guaranteeing safe driving. By taking any other actions that may lead to an increase in HOs, collisions, or traffic violations, the AV receives a penalty. We define the transportation reward as [24]:

$$r_t^{\text{j,tran}} = c_1 \left(\frac{v_t^j - v_{\min}}{v_{\max} - v_{\min}} \right) - c_2 \cdot \delta_2 + c_3 \cdot \delta_3 + c_4 \cdot \delta_4, \quad (16)$$

where $v_t^j, v_{\min}$ and $v_{\max}$ are the current longitudinal velocity for AV j on time t, the minimum and maximum speed limits, and $\delta_2, \delta_3, \delta_4$ is the collision indicator, AV right lane indicator, on road indicator, respectively. c_1 and c_2 are the weights that adjust the value of the AV transportation reward with its collision penalty. c_1 indicates that the reward received when driving at full speed, linearly mapped to zero for lower speeds. c_3 shows that AV was rewarded for driving on the right-most lanes, and linearly mapped to zero for other lanes. c_4 is the on-road reward factor, which penalize the AV for driving off highway. It is important to note that negative rewards are not allowed since they might encourage the agent to prioritize ending an episode early by causing a collision instead of taking the risk of receiving a negative return if no satisfactory trajectory is available.

For the telecommunication side, we define the reward for AV j associated with BS i^* at time step t as:

$$r_t^{\text{j,tele}} = c_5 \text{WR}_{i^*,j,t} \left(1 - \min(1, \xi_t^j)\right), \quad (17)$$

where c_5 is the coefficient to scalarlize weighted datarate. $\text{WR}_{i^*,j,t}$ is the achievable data rate compute by (6) and ξ_t^j is the HO probability of AV j computed by dividing the number of HOs accounted until the current time t by the time duration of previous time slots in the episode[3].

Based on the instantaneous reward, we compute the accumulated rewards, which is the summation of discounted reward among all target AVs on the highway in each training episode. The expected return is defined as follows:

$$\mathbf{Q}_\pi(s, a, \boldsymbol{\omega}) = \mathbb{E}_\pi \left[\sum_{j=1}^{M_1} r_t^{\text{j,tran}}, \sum_{j=1}^{M_1} r_t^{\text{j,tele}} \right] \quad (18)$$

Our MOMDP optimal strategy for maximizing the expected reward involves the simultaneous maximization of both transportation and telecommunications objectives, i.e., $\max_\pi \mathbf{Q}_\pi(s, a, \boldsymbol{\omega})$.

IV. Multi-Policy Envelope MORL Algorithm

In contrast to conventional DRL, MORL requires the agent to optimize multiple objectives simultaneously. These objectives might have predefined preferences, or the preferences could be unknown. A multiple-policy envelope solution for MORL is proposed in Section IV for unknown preferences. The single policy solutions to the MORL problem with predefined preferences are discussed in **Appendix**. Although single-policy methods are adequate when we possess prior knowledge of task preferences, the acquired policy is constrained in its adaptability to situations with varying preferences. For instance, collision avoidance may not remain a high priority in highway environments with reduced vehicle density. Also, in traffic jams or parking lots where AVs are still, the preference for telecommunication rewards becomes higher.

Our approach separates the model training and evaluation phases. The MORL agent is trained offline using a predefined simulated environment. During the evaluation stage, we use a *pool* of pretrained MORL models, each tailored to a predefined configuration defined by a given number of AVs (representing traffic conditions), number of RF/THz BSs, and a specific topology. This approach is also adopted by 3GPP [36], i.e., at run time, the controller simply selects the MORL model

[3]Note that $c_1 \dots c_5$ are the weights to set the priority of each term. For instance, c_2 needs to be sufficiently large compared to other coefficients for collision avoidance. The highest penalty applies to vehicle collision.

that best matches the observed snapshot, achieving zero online training overhead [36].

In contrast to single-policy methods, multi-policy MORL methods optimize different objectives simultaneously by maximizing a vector of rewards associated with these objectives. Our proposed MORL framework reduces reliance on predefined preferences and scalar reward combinations, enabling dynamic adjustment to associated tasks featuring distinct preferences. This approach is effective in identifying Pareto-optimal policies when preferences are unknown.

Our proposed MO-DDQN-Envelope algorithm is designed to learn a spectrum of Pareto-optimal policies simultaneously within a preference space Ω, as described in Section III-A and illustrated in Fig. 3. Unlike the Envelope-MOQ model proposed in [35], which employs the REINFORCE algorithm, our MO-DDQN-Envelope algorithm incorporates DDQN to enhance both convergence stability and sample training efficiency. REINFORCE, as a policy gradient method, often suffers from high variance in complex environments like *RF-Thz-Highway-Env*, leading to unstable updates. In contrast, DDQN leverages temporal difference (TD) learning, which reduces variance and mitigates the overestimation bias. Furthermore, DDQN employs replay experience to enable the agent to learn from past experiences multiple times, improving sample efficiency, and uses target networks to stabilize learning by ensuring consistent targets during updates.

During each time step, observation information is captured in the *RF-THz-Highway* environment. From this observation, the tuple $\{s_t, s_{t+1}, \boldsymbol{\omega}_t\}$ is computed. Following states information acquisition, the hindsight experience replay (HER) technique is employed to sample preference weights from the replay preference pool $\mathcal{D}_\mathcal{T}$. Then, homotopy optimization is applied to execute gradient descent, as indicated in (24). Subsequently, we perform Q network clone from evaluation network to target network periodically for every N^- steps. Notably, unlike prior single policy MORL approaches that scalarize rewards before the experience replay, MO-DDQN-envelope scalarizes rewards after gradient descent. We elaborate on the Bellman operator update phase, the HER phase and homotopy optimization phase in detail in what follows.

1) Bellman Operation with Optimal Filter: In the context described in Section III-A and referenced by [35], the expected discounted return under a policy π is defined as $\mathbf{Q}_\pi(s, a, \boldsymbol{\omega}) = \mathbb{E}_\pi\left[\mathbf{r}(s_t, a_t) + \gamma \mathbf{Q}_\pi(s_{t-1}, a_{t-1}, \boldsymbol{\omega})\right]$. Yang *et al.* in [35] further introduces the concept of an optimal filter $\mathcal{H}$[4], which is applied to $\mathbf{Q}_\pi(s, a, \boldsymbol{\omega})$ to obtain $(\mathcal{H}\mathbf{Q})_\pi(s, a, \boldsymbol{\omega}) = \arg_Q \sup_{a \in \mathcal{A}, \boldsymbol{\omega}' \in \Omega} \mathbf{Q}_\pi(s, a, \boldsymbol{\omega}')$. The $\arg_Q$ represents a multi-objective supremum value, ensuring that $(a, \boldsymbol{\omega}')$ achieves the maximum supremum across actions in space $\mathcal{A}$ and states $\boldsymbol{\omega}'$ within the space Ω. Consequently, we utilize (13) to focus the optimization on actions solely dependent on $\mathcal{H}$. The MO optimality operator can thus be defined as:

$$\mathbf{Q}(s, a, \boldsymbol{\omega}) = \mathbb{E}_{s_{t+1}}\left[\mathbf{r}(s_t, a_t) + \gamma(\mathcal{H}\mathbf{Q})(s_{t+1}, \boldsymbol{\omega})\right] \tag{19}$$

[4]The optimal filter $\mathcal{H}$ is instrumental in solving the convex envelope of PPF, which represents the current solution frontier. This process is key in optimizing the Q-function, $\mathbf{Q}_\pi$ for a given state s and preference weights $\boldsymbol{\omega}$.

During the training phase, $\boldsymbol{\omega}$ values are collected through free exploration of the environment. As can be seen in Fig. 2, ω_i specifies preference weights to balance objectives. In the initial free exploration phase, ω_i uniformly samples from the sampling space Ω. Uniform sampling ensures that the exploration covers a broad range of trade-offs, increasing the likelihood of finding Pareto-optimal solutions across the entire objective space.

Unlike scalarized single-policy approaches discussed in **Appendix**, which fail to adapt the scalar utility across varying $\boldsymbol{\omega}$, the convex envelope formulation explicitly leverages the supremum operator to optimize across all possible actions a and preference vectors $\boldsymbol{\omega}'$.

2) Hindsight Experience Replay (HER): HER is a method to train a RL agent to achieve multiple preferences to serve multiple objectives [35], [37]. The RL agent follows a policy based on a randomly selected goal in each episode and uses the previous trajectory to update other goals simultaneously.

In our enhanced MO-DDQN-envelope network, leveraging HER, we employ the sampling process from two distinct replay pools $\mathcal{D}_\mathcal{T}$ and $\mathcal{D}_\omega$, targeting both transition mini-batches and preference vectors. We extract $N_\mathcal{T}$ mini-batch transitions,$(s_z, a_z, \mathbf{r}_z, s_{z+1})$ to form replay buffer pool $\mathcal{D}_\mathcal{T}$, such as $(s_z, a_z, \mathbf{r}_z, s_{z+1}) \sim \mathcal{D}_\mathcal{T}$, where $z \in [1, N_\mathcal{T}]$. Concurrently, we sample preference vectors ω_g in $\mathcal{D}_\omega$ to form replay buffer $\mathcal{W} \equiv \{\omega_g \sim \mathcal{D}_\omega\}$, with $g \in [1, N_\omega]$, N_ω indicates the count of preference weights in $\mathcal{W}$. Therefore, the agent AV can replay the trajectories with any preferences using "hindsight" since preferences only impact agent AVs actions rather than highway environment dynamics [35].

3) Homotopy Optimization: Our goal is to generate a single model which adapts the entire pareto frontier space Ω. By sampling $N_\mathcal{T}$ transitions $(s_z, a_z, \mathbf{r}_z, s_{z+1})$ and N_ω preference weights $\mathcal{W} = \{\omega_g \sim \mathcal{D}_\omega\}$ in respective replay buffer $\mathcal{D}_\tau$ and $\mathcal{D}_\omega$, we define MO-DDQN-envelope element-wise target function [35] as follows:

$$\hat{\mathbf{Q}}(s_z, a_z, \mathbf{r}_z, s_{z+1}, \omega_g) = \mathbf{r}_z + \gamma \max_{a' \in \mathcal{A}, \boldsymbol{\omega}' \in \mathcal{W}} [\omega_g]^T \mathbf{Q}(s_{z+1}, a', \boldsymbol{\omega}') \tag{20}$$

for $\forall z \in [1, N_\mathcal{T}]$ and $\forall g \in [1, N_\omega]$. Finding the optimal preference weight $\boldsymbol{\omega}'$ in Ω can be an NP-hard problem due to the size and complexity of Ω. Instead, finding the optimal preference $\boldsymbol{\omega}'$ in $\mathcal{W}$ is feasible. In contrast to Ω, $\mathcal{W}$ contains only those preferences that align with optimal solutions. This is because $\mathcal{W}$ corresponds to the convex boundary of the Pareto frontier, where the utility projection $\boldsymbol{\omega}^\top \mathbf{Q}(s, a, \boldsymbol{\omega}')$ achieves its maximum for some preference $\boldsymbol{\omega}' \in \Omega$. Consequently, $\mathcal{W}$ serves as a reduced yet comprehensive subset of Ω, facilitating efficient exploration and optimization without excluding any optimal solutions. From a computational perspective, the use of $\mathcal{W}$ simplifies the optimization process. By maintaining the envelope $\sup_{\boldsymbol{\omega}'} \boldsymbol{\omega}^\top \mathbf{Q}(s, a, \boldsymbol{\omega}')$, the algorithm effectively focuses on preferences that yield maximal utility, avoiding unnecessary evaluations in suboptimal regions of Ω. This ensures that the updated policies and preferences are always aligned with the optimal frontier. Furthermore, the iterative refinement of $\mathcal{W}$ through envelope updates enables the model to integrate information from previously explored trajectories

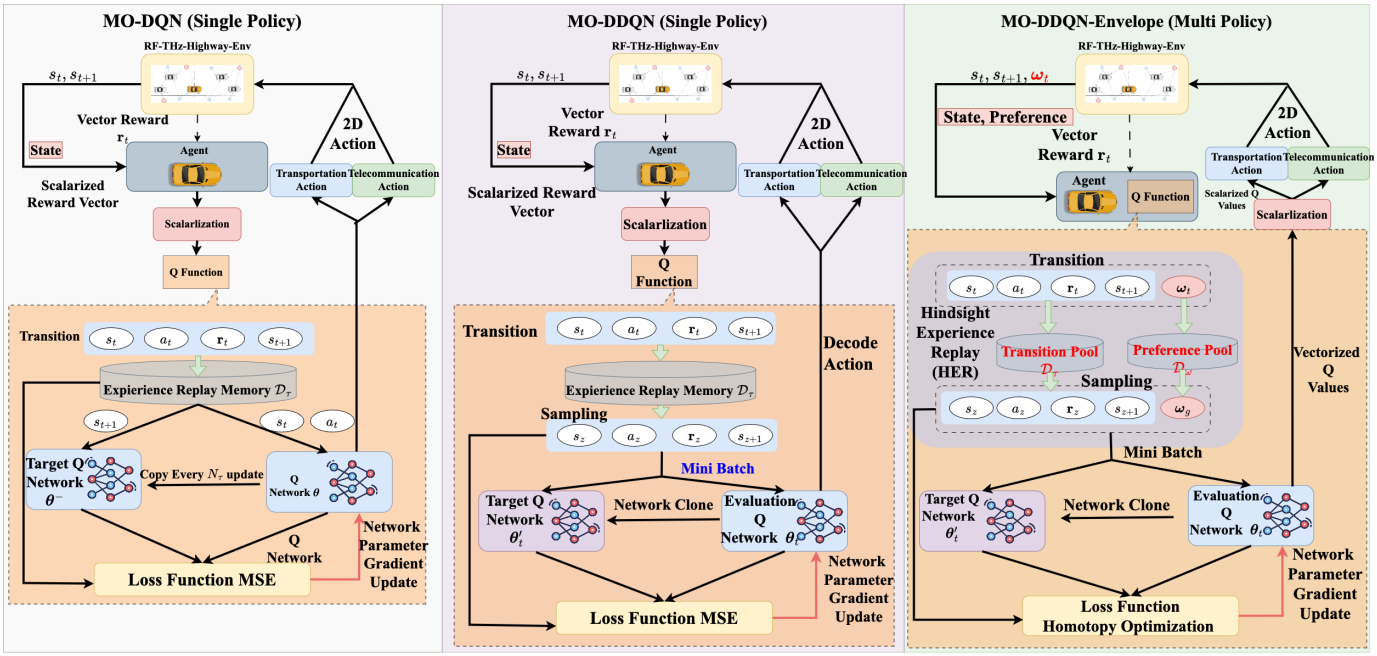


Figure 3: Comparison of MO-DQN, MO-DDQN, and the proposed MO-DDQN-envelope framework

stored in the preference pool D_ω. This process results in faster convergence and improved sample efficiency compared to directly exploring the entire space Ω. By focusing optimization within $\mathcal{W}$, the approach avoids unnecessary evaluations in suboptimal regions of Ω, thereby improving computational efficiency.

By replay sampling transition $(s_t, a_t, \mathbf{r}_t, s_{t+1})$ across $N_\mathcal{T}$ transitions, we acquire the empirical estimate target function over new state s_{t+1} as:

$$\hat{\mathbf{Q}}(s_t, a_t, \boldsymbol{\omega}_t; \boldsymbol{\theta}'_t) = \mathbb{E}_{s_{t+1}}[\mathbf{r}_t + \gamma \arg_Q \max_{a_t, \boldsymbol{\omega}'_t} \boldsymbol{\omega}^T \mathbf{Q}(s_{t+1}, a_t, \boldsymbol{\omega}'_t; \boldsymbol{\theta}'_t)] \tag{21}$$

where $\mathbf{Q}(\cdot)$ revisits (19). To ensure the correctness of the training for the target value $\hat{\mathbf{Q}}$, which should be as close as possible to the actual value ($\mathbf{Q}$). The loss function $\mathcal{L}^A(\boldsymbol{\theta}_t)$ in each time step t is defined as:

$$\mathcal{L}^A(\boldsymbol{\theta}_t) = \mathbb{E}_{s_t, a_t, \omega_t}\left[||\hat{\mathbf{Q}}(s_t, a_t, \omega_t; \boldsymbol{\theta}'_t) - \mathbf{Q}(s_t, a_t, \omega_t; \boldsymbol{\theta}_t)||_2^2\right] \tag{22}$$

Since $\mathcal{L}^A(\boldsymbol{\theta}_t)$ contains many local maxima and minima, it is difficult to find the mean square error (MSE) and hard to optimize Q_θ. To smooth the landscape of loss function $\mathcal{L}^A(\boldsymbol{\theta}_t)$, we introduce the auxiliary loss function $\mathcal{L}^B(\boldsymbol{\theta}_t)$ as:

$$\mathcal{L}^B(\boldsymbol{\theta}_t) = \mathbb{E}_{s_t, a_t, \omega_t}\left[|\omega_t^T \hat{\mathbf{Q}}(s_t, a_t, \omega_t; \boldsymbol{\theta}'_t) - \omega_t^T \mathbf{Q}(s_t, a_t, \omega_t; \boldsymbol{\theta}_t)|\right] \tag{23}$$

$\mathcal{L}^B(\boldsymbol{\theta}_t)$ contributes smooth policy adaptation for enhancing training efficiency with fewer spikes. $\mathcal{L}^B(\boldsymbol{\theta}_t)$ is advantageous for boosting agent training, but not as good for accurate approximation as $\mathcal{L}^A(\boldsymbol{\theta}_t)$ [35]. Both $\mathcal{L}^A(\boldsymbol{\theta}_t)$ and $\mathcal{L}^B(\boldsymbol{\theta}_t)$ are averaged over ω_t which highlights the sampling preference feature in the proposed algorithm. However, specific weight ω_t in the past training is not directly applied to the target state-action transitions. The proposed MO-DDQN-envelope can reevaluate past transitions in $\mathcal{D}_\mathcal{T}$ with later new preferences to enhance learning efficiency and sample utilization.

Combining (22) and (23), we generate loss function

$$\mathcal{L}(\boldsymbol{\theta}_t) = (1 - \lambda_t)\mathcal{L}^A(\boldsymbol{\theta}_t) + \lambda_t \mathcal{L}^B(\boldsymbol{\theta}_t) \tag{24}$$

We progressively increase the value of λ_t from 0 to 1 throughout the training process, thereby transitioning the loss function from $\mathcal{L}^A(\boldsymbol{\theta}_t)$ to $\mathcal{L}^B(\boldsymbol{\theta}_t)$. During training, λ_t continuously evolves within the range $0 < \lambda_t < 1$, ensuring a gradual shift that balances the contributions of the two objectives as the training advances [35], [38]. This method called *homotopy optimization* [38] is effective since for each update step, it uses the optimization result from the previous step as the initial guess. In the envelope deep MORL algorithm, $\mathcal{L}^A(\boldsymbol{\theta}_t)$ first ensures the prediction of Q is close to any real expected total reward, though not necessarily optimal. And, then $\mathcal{L}^B(\boldsymbol{\theta}_t)$ can pull the current guess along the direction with better utility. As depicted in Figure 4, the MSE loss $\mathcal{L}^A(\boldsymbol{\theta}_t)$ is difficult to optimize since there are many local minima over its landscape. Although the loss of the value metric $\mathcal{L}^B(\boldsymbol{\theta}_t)$ has fewer local minima, it is also difficult for optimization since there are many vectors Q minimizing value metric $\omega_t^T \cdot |\hat{\mathbf{Q}}(s_t, a_t, \omega_t; \boldsymbol{\theta}'_t) - \mathbf{Q}(s_t, a_t, \omega_t; \boldsymbol{\theta}_t)|$, making $\mathcal{L}^B(\boldsymbol{\theta}_t)$ is flat. The homotopy path connecting $\mathcal{L}^A(\boldsymbol{\theta}_t)$ and $\mathcal{L}^B(\boldsymbol{\theta}_t)$ provide better opportunities to find the optimal global parameters $\mathcal{L}(\boldsymbol{\theta}_t)$.

We first trying to reduce the discrepancy between target and estimate $\mathbf{Q}$ value as $(\hat{\mathbf{Q}}(s_t, a_t, \omega_t; \boldsymbol{\theta}_t) - \mathbf{Q}(s_t, a_t, \omega_t; \boldsymbol{\theta}_t)$ and then taking gradient descent $\nabla_{\boldsymbol{\theta}_t}$ for estimate $\mathbf{Q}$ value to adjust direction to reduce MSE. Consequently, parameter for MO-DDQN-envelope will be updated as,

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + \mathbb{E}_{s_t, a_t, s_{t+1}}[(\hat{\mathbf{Q}}(s_t, a_t, \omega_t; \boldsymbol{\theta}_t) - \mathbf{Q}(s_t, a_t, \omega_t; \boldsymbol{\theta}_t))^T \nabla_{\boldsymbol{\theta}_t} \mathbf{Q}(s_t, a_t, \omega_t; \boldsymbol{\theta}_t)] \tag{25}$$

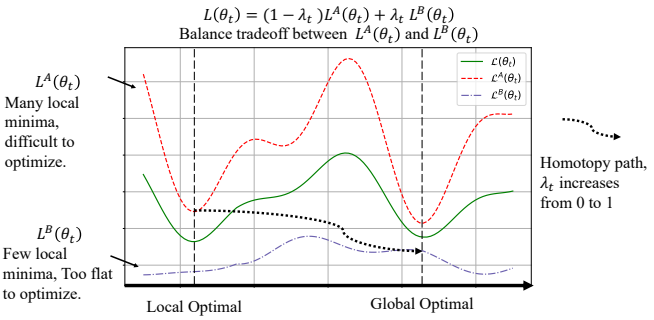


Figure 4: An explanation for homotopy optimization method used in the MO-DDQN-Envelope

We reset our target Q-network with evaluation Q-network every N^- steps, i.e., $\theta \leftarrow \theta'$. The training algorithm of the proposed MO-DDQN-envelope is shown in **Algorithm** 1.

A. Complexity Analysis

1) Training Complexity Analysis: We first analyze the main loop of **Algorithm** 1. The key components contributing to the time complexity include:

- *Action Selection:* We consider the horizon limit for each episode, denoted as T_{hl}. The process of selecting an action at each timestep incurs a time complexity of $\mathcal{O}(|\mathcal{A}|)$ for each target AV, where $|\mathcal{A}|$ represents the size of the action space. Thus, the time complexity is $\mathcal{O}(M_1 \cdot T_{\text{hl}} \cdot |\mathcal{A}|)$.
- *Hindsight Experience Replay (HER):* Sampling $N_{\mathcal{T}}$ transitions from the HER replay buffer and N_ω preferences results in a time complexity of $\mathcal{O}(N_{\mathcal{T}} \cdot N_\omega)$.
- *Homotopy Optimization:* This phase involves a Fully Connected Neural Network (FCNN) consisting of an input layer, an output layer, and E fully connected hidden layers. Let N_e denote the number of neurons in the e-th fully connected layer. The time complexity for this phase is $\mathcal{O}\left(\sum_{e=1}^{E+1}(N_{e-1} \cdot N_e)\right)$, accounting for the computational cost for each layer's connections.
- *Policy Adaptation:* Utilizing a DDQN, the input layer's neurons correspond to the dimensionality of the state space $\mathcal{S}$, and the output layer's neurons correspond to the action space $\mathcal{A}$. Thus, the numbers of neurons in the input and output layers are $N_0 = |\mathcal{S}|$ and $N_{E+1} = |\mathcal{A}|$, respectively. The time complexity for this phase is $\mathcal{O}(|\mathcal{S}| \cdot |\mathcal{A}|)$, which can be considered constant upon convergence.

Therefore, the overall training complexity for MO-DDQN-envelope on 1 episode can be expressed as $\mathcal{O}(M_1 \cdot T_{\text{hl}} \cdot |\mathcal{A}| + N_{\mathcal{T}} \cdot N_\omega + \sum_{e=1}^{E+1}(N_{e-1} \cdot N_e) + |\mathcal{S}| \cdot |\mathcal{A}|)$.

2) Evaluation Complexity Analysis: Assume we have N_l layers in the neural network, each containing N_u neurons. According to Section III-B, the neural network processes $|\mathcal{S}|$ inputs from the state space and produces $|\mathcal{A}|$ possible actions as output. The overall complexity for a forward pass through the network is given by:

$$\mathcal{O}(|\mathcal{S}| + |\mathcal{A}| + N_u^{N_l}) \approx \mathcal{O}(N_u^{N_l}),$$

and $N_u^{N_l}$, the computation for the hidden layers, dominates the complexity. For evaluation, we assume a single step is implemented with a constant-time complexity of $\mathcal{O}(1)$. If there are M_1 target AVs and each travels for at most T_{hl} time steps in one episode, the total number of steps in an episode is $M_1 \cdot T_{hl}$. Thus, the overall complexity for one simulation episode

Algorithm 1: Multi-Objective Envelope DDQN

Result: Learned action-value function $\mathbf{Q}_\theta$ and Policy π
Data: Evaluation Q-network $\mathbf{Q}_\theta$, Target Q-network $\mathbf{Q}_{\theta'}$, Preference sampling pool $\mathcal{D}_\omega$, HER transition sampling pool $\mathcal{D}_{\mathcal{T}}$, Balance weight path p_λ
Initialization:
HER replay buffer $\mathcal{D}_{\mathcal{T}} \leftarrow \emptyset$,
Initialize Q-network weights θ randomly,
Initialize target Q-network weights $\theta' \leftarrow \theta$,
Initialize $Q(s,a)$ for all states s and actions a, including AVs, TBSs, RBSs.
while *episode* $<$ *episode limit and runtime* $<$ *time limit* **do**
  Initialize $t \leftarrow 0$ and state s_t based on environment
  while $t \leq T_{hl}$ **do**
    for *Target AV* j ***from*** *1* ***to*** M_1 **do**
      a_t select action from $\mathcal{A}$ with probability of ϵ or Select a_t from $a_t = \arg\max_{a \in \mathcal{A}} \boldsymbol{\omega}^T \mathbf{Q}(s_t, a, \boldsymbol{\omega}; \boldsymbol{\theta}_t)$ with probability of $1-\epsilon$.
      Derive a_t^{tran} and a_t^{tele} from a_t;
      Apply a_t^{tran} and a_t^{tele} to target AV j;
      Observe vector reward $\mathbf{r}_t$ and next state s_{t+1};
    end
    if *update neural network* **then**
      Store $(s_t, a_t, \mathbf{r}_t, s_{t+1})$ in $\mathcal{D}_{\mathcal{T}}$;
      Hindsight Experience Replay (HER):
      $\{(s_z, a_z, \mathbf{r}_z, s_{z+1}) \sim \mathcal{D}_{\mathcal{T}}\}$;
      Sample N_ω preferences $\mathcal{W} = \{\omega_g \sim \mathcal{D}_\omega\}$;
      Bellman Update:
      Compute $\hat{\mathbf{Q}}(s_z, a_z, \mathbf{r}_z, s_{z+1}, \omega_g)$ for each sampled transition and preference:

$$\begin{cases} \mathbf{r}_z, & \text{if } s_{z+1} \text{ is terminal} \\ (20), & \text{otherwise} \end{cases}$$

      $\forall z \in [1, N_{\mathcal{T}}]$ and $\forall g \in [1, N_\omega]$
      Homotopy Optimization:
      Update $\mathbf{Q}_\theta$ by minimizing the loss with gradient descent by (24);
      Gradually increase λ following the path p_λ;
    end
    Update target $\mathbf{Q}_{\theta'}$ weights $\theta' \leftarrow \theta$ by (25) every N^- steps;
  end
  $t \leftarrow t+1$
  Compute policy π based on learned $\mathbf{Q}_\theta$;
end

is:

$$\mathcal{O}(M_1 \cdot T_{hl} \cdot (N_u^{N_l} + \mathcal{O}(1))) \approx \mathcal{O}(M_1 \cdot T_{hl} \cdot N_u^{N_l}),$$

where the constant $\mathcal{O}(1)$ is negligible compared to the forward pass complexity.

V. Simulation and Performance Evaluation

In this section, we demonstrate the performance of the proposed algorithms and highlight the complex dynamics between wireless connectivity, traffic flow, and AV's speed. All Experiments are executed on a PC equipped with Windows 11, Intel i7-8770 CPU 3.2 GHz and 16 GB DDR5, AMD RX580 8GB GDDR5. Additionally, *Google Colab*, is the cloud platform employed for reproduction and verification.

A. Simulation Environment

Our proposed simulation framework is composed of three main components:

- **Telecommunication and Transportation Environment:** We introduce the *MO-RF-THz-Highway-env* framework[5], an enhanced version of *highway-env* [24], designed to support both autonomous driving policy and 5G/6G network selection for multiple AVs.
- **Extended MO-Gymnasium:** We extend *MO-Gymnasium*[6] [39] to provide an application programming interface (API) to communicate between DRL algorithms and *MO-RF-THz-Highway-env*.
- **MORL Algorithms Simulation:** For single policy MORL, simulation utilizes modified *rl-agents*[7] from [40]. For multi-policy MORL, the proposed MO-DDQN-envelope[8] and the other multi-policy algorithms simulation are extended from *MORL-baselines* in [41].

The *MO-RF-THz-Highway-env* features five one-way lanes, each with a length of 1500m and a width of 4m, as depicted in Figure 1. In our experimental setup, a default configuration consists of 5 target AVs and 20 surrounding AVs, each having a length of 5m. The longitudinal velocity of each AV ranges from 10 m/s to 35 m/s. At the beginning of each episode, these AVs are randomly placed across the five lanes. Along both sides of the highway, 5 RBSs and 10 to 50 TBSs are also randomly positioned to ensure a non-uniform distribution at the beginning of each episode. This random placement strategy aims to facilitate the examination of MORL training effectiveness across various VNets and traffic scenarios, as opposed to a singular, common scenario. The maximum duration for each episode is set to 30 time steps. An episode is considered *collision-free* if it meets two criteria: absence of collisions among all target AVs, and maintenance of a high weighted data rate regarding (6) during travel through the episode. For single policy MORL, we set rewards coefficients in (16) (17) from Section III-D, c_1, c_2, c_3, c_4, c_5 set to $0.4, 1, 0.1, 0.2, 4.5 \times 10^{-7}$, respectively.

[5]https://github.com/sunnyyzj/highway-env-1.7
[6]https://github.com/sunnyyzj/MO-Gymnasium
[7]https://github.com/sunnyyzj/rl-agents
[8]https://github.com/sunnyyzj/morl-baselines

Parameter	Value
Value used in system model	
RBSs frequency (f_R)	3.5 GHz
TBSs frequency (f_T)	1 THz
Maximum number of affordable AVs quota for a single RBS, TBS (Q_R), (Q_T)	5, 10
Antenna gain for TBSs and RBSs (G_T^{tx}), (G_T^{rx})	316.2
RBSs channel bandwidth (W_R)	4×10^7
TBSs channel bandwidth (W_T)	5×10^8
Transmit powers of RBSs and TBSs (P_R^{tx}), (P_T^{tx})	1 W
Molecular absorption coefficient (K_a)	0.05 m^{-1}
Path loss exponent (α)	4
Length for each AV (l_j)	5 m
Target AV heading and lateral control gain (K_j^{ψ}),(K_j^{y})	5, $\frac{5}{3}$
Maximum AV steering angle for AV j ($\max \beta_j$)	$\frac{\pi}{3}$
Surrounded AV j desired maximum acceleration and deceleration (a_j)	3 m/s , -5 m/s
Acceleration reduction factor (δ_a)	4
Number of lanes (N_L)	5
Value used in MORL	
Learning rate (α_l)	3×10^{-4}
Discount factor (γ)	0.995
Size of the hidden layers of the value NN	$[256, 256, 256, 256]$
Epsilon decay parameter (ϵ)	0.1
MO-DDQN-envelope HER transition pool size ($N_{\mathcal{T}}$)	2×10^6
The number of weight vectors to sample for the envelope target (N_ω)	4
Frequency for cloning evaluation to target network (N^-)	200
Episode horizon limit T_{hl}	30

Table II: Values of system parameters in experiments

The collision coefficient c_2 is set significantly higher than the others because collision avoidance is our highest-priority and incurs the greatest penalty [34], [42]. On the other hand, the achievable handover-aware data rate ranges from 5×10^7 bits per second (bit/s) to 5×10^8 bit/s. To balance the transportation and telecommunication objectives effectively, c_5 is set to 4.5×10^{-7}, ensuring that the data rate is appropriately scaled for optimizing weighted-sum single policy MORL.

We utilize a Multi-Layer Perceptron (MLP) to construct the Q_θ neural network for the training and evaluation phases. The architecture consists of three fully connected layers (FNNs), each containing 128 neurons. Following these layers, we apply the ReLU activation function to improve training efficiency. The output layer of the target policy network Q'_θ employs a sigmoid activation function to constrain the output range of actions. This ensures that the action values remain within a feasible and interpretable range, which is critical for MOO decision-making tasks. This architecture was chosen after a comprehensive hyperparameter tuning process. This configuration strikes a balance between model accuracy, training efficiency, and convergence performance. Simulation parameters are detailed in Table II unless otherwise specified.

We evaluated all methods across five test instances, varying key variables: desired minimum and maximum longitudinal velocities ($v_{\text{min}}, v_{\text{max}}$), the number of TBSs ($n_T$), and the number of AVs (M). These combinations are given in Table III.

B. Baselines and Evaluation Metrics

To comprehensively evaluate the performance of MO-DDQN-envelope, we implement the following baseline al-

Instance	$v_{\min}$	$v_{\max}$	n_T	M
I-(20,30,20,20)	20 m/s	30 m/s	20	20
I-(25,35,20,20)	25 m/s	35 m/s	20	20
I-(20,30,10,20)	20 m/s	30 m/s	10	20
I-(20,30,20,50)	20 m/s	30 m/s	20	50
I-(30,40,20,20)	30 m/s	40 m/s	20	20

Table III: Test Instances with Parameters.

gorithms for comparison: MO-DQN, MO-DDQN (as discussed in **Appendix**), MO-dueling-DDQN, and MO-PPO. MO-dueling-DDQN and MO-PPO are variations of MO-DDQN, utilizing single-policy multi-objective dueling DDQN and Proximal Policy Optimization (PPO) algorithms, respectively, instead of DDQN. These algorithms, specifically developed for performance evaluation, are extensions of single-policy RL methods originally proposed in [43] and [44], respectively. Additionally, we incorporate several state-of-the-art algorithms from MORL-Baselines [41] for comparison. These include Scalarized Q-learning for single-policy MORL (MO-Q) [45], Scalarized DQN for single-policy DQN (MO-DQN) [1], Action Branching Architectures with Dueling DDQN for MORL (MO-Dueling-DDQN) [12], Multi-objective exploration for proximal policy optimization (MO-PPO) [46], and Expected Utility Policy Gradient (EUPG) [47]. To evaluate the performance, we consider the following metrics. Assume episode e ends on time step T_e. we define **(1)** total transportation reward: $R_e^{tran} = \mathbb{E}_{j\in M_1}[\sum_{t=1}^{T_e} r_t^{\mathrm{j,tran}}]$. **(2)** total telecommunication reward: $R_e^{tele} = \mathbb{E}_{j\in M_1}[\sum_{t=1}^{T_e} r_t^{\mathrm{j,tele}}]$. **(3)** collision rate: $\delta_e = 1 - \frac{T_e}{T_{hl}}$ **(4)** HOs Probability: $\xi_e = \mathbb{E}_{j\in M_1}[\xi_{T_e}^j]$. where $r_t^{\mathrm{j,tran}}$ and $r_t^{\mathrm{j,tele}}$ are instantaneous transportation and telecommunication rewards specified in equations (16) and (17), respectively. T_{hl} represents the horizon limit of each episode. $\xi_{T_e}^j$ denotes HOs probability by the end of each episode (T_e's step) defined in equation (17). Table IV evaluate total transportation reward (16), total telecommunication reward (17), total rewards defined by $R_e^{total} = R_e^{tele} + R_e^{tran}$ over MO-DQN, MO-DDQN, and MO-DDQN-Envelope.

C. Results and Discussions

1) Training Performance: We examine the training performance of the proposed MO-DDQN-envelope algorithm and compare it with other baselines of MORL approaches. Fig. 5 depicts total transportation rewards, total telecommunication reward, collision Rate, and HOs probability as a function of desired velocity of AV. MO-DDQN-envelope performs better than benchmark algorithms (i.e. MO-DQN and MO-DDQN) when evaluating performance over every 100 episodes. Shown by learning curves, we note that MO-DDQN-envelope has slower convergence to higher total cumulative training rewards no matter transportation and telecommunication sides. However, MO-DDQN-envelope algorithm balances both transportation and telecommunication objectives. Collision rate and HOs probability reduce better than the other benchmarks. To better understand this improvement, recall MO-DDQN-envelope samples experience from the replay buffer which contains recent past preferences and rarely new exploration preferences. Past preferences are based on the weight vector $\vec{\omega}$ in terms of transportation and telecommunication objectives $\vec{\omega} = [\omega_{\mathbf{tran}}, \omega_{\mathbf{tele}}]^T$ which marginally improves the training for each objective individually regardless of preferences.

For the convergence rate, thanks to *Homotopy Optimization*, training first focuses on training accuracy on two objectives but later gradually focuses on faster convergence, training rewards are less viable after 3000 episodes of training than in the early stage. Also, we found the collision rate is significantly reduced from 0.7 to around 0.2, which also illustrates the training model is improving safety on the highway.

2) Impact of BSs Density: We evaluate the performance by averaging over 500 evaluation epochs on models after 4500 episodes of training. In each evaluation step, we randomly distribute different numbers of TBSs alongside the highway in the simulation environment. As depicted in Fig. 6, MO-DDQN-envelope gains an evaluation advantage compared to other benchmarks. As the number of BSs grow, the transportation rewards do not change significantly, however, the telecom rewards first increases due to better connectivity and later decreases due to more HOs. Growing TBSs also increases the average collision rate due to potential reduction in AVs speed to maintain connectivity.

3) Impact of the Number of AVs: As shown in Fig. 6, increasing the number of AVs leads to more crowded highway scenarios. Thus, more grouped AVs connect to the same BS resulting in network outages due to the maximum quota at each BS. Also, frequent lane changes and speeding on the crowded highway cause more congestion and collisions, which reduces transportation performance.

4) Impact of Desired AV Speeds: Fig. 6 depicts that slow moving AVs outperform in terms of both transportation and telecommunications. Increasing speeds lead to higher collision occurrences. Moreover, AVs at higher speeds switch BSs more frequently, incurring significant handover penalties according to (17), thus adversely affecting rewards in both domains.

5) Pareto Front Analysis: We employ the CCS in (15) as a means to assess the excellence of the estimated Pareto fronts. A greater Pareto frontier value indicates a closer proximity of the Pareto front to the optimal one in terms of transportation and telecommunication objectives. To compute CCS, we select performance on single policy MO-DQN and MO-DDQN as reference points. Recall we need to maximize both transportation rewards and telecommunication rewards. The best solutions are situated in the top right corner, as depicted in Fig. 7. Specifically, it demonstrates that our proposed algorithm MO-DDQN-envelope outperforms the MORL baselines in approximating the Pareto fronts. However, in the high-density transportation scenario, MO-DDQN yields a Pareto front of similar quality to the other baselines.

6) Training and Evaluation Time Complexity Per Step : We trained each algorithm for 4000 episodes, with up to T_{hl} steps per episode, resulting in a total duration per step of approximately 0.643 seconds. The breakdown of time per step during training and evaluation is presented in Table V. The reward objectives are modified as follows:

- **Telecommunication Only:** This scenario considers only the telecommunication reward in the MDP design. The

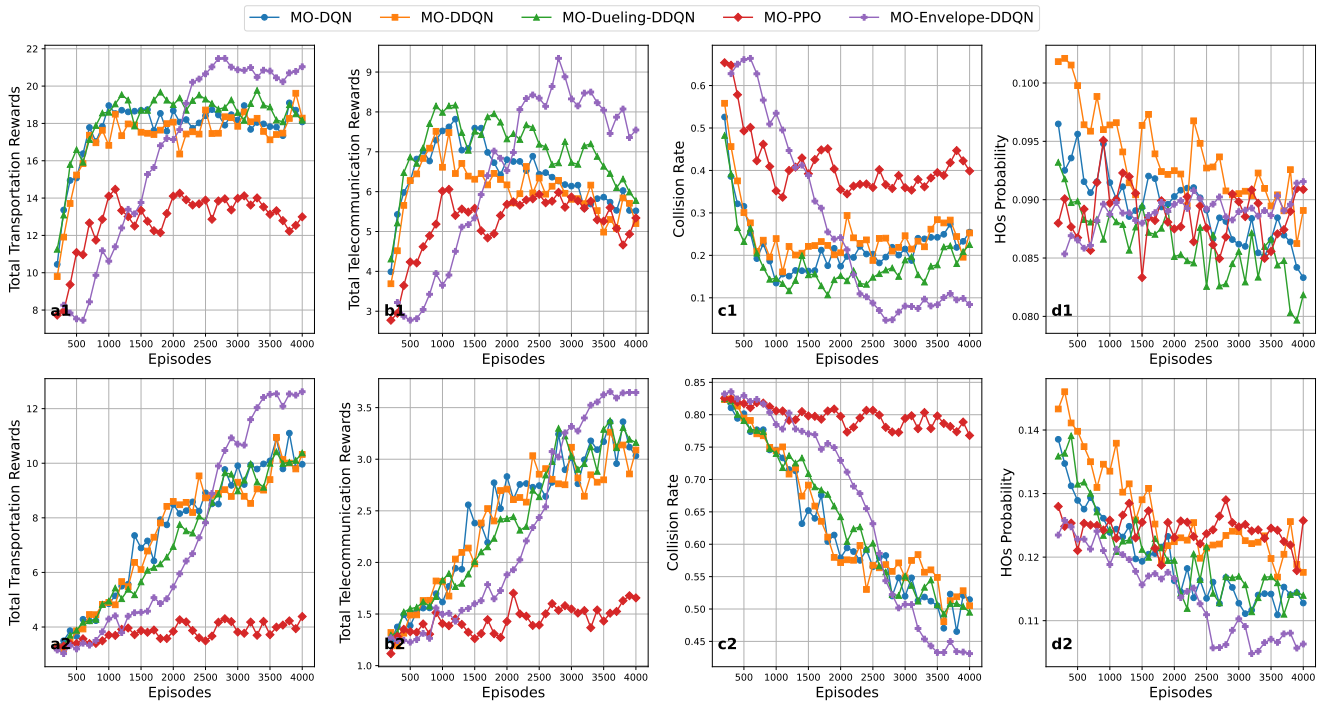


Figure 5: Training performance on (a) total transportation rewards, (b) total telecommunication rewards, (c) collision rate, and (d) HOs probability. $n_T = 20$, the desired minimum and maximum longitudinal velocities are $v_{\min} = 20$m/s and $v_{\max} = 30$m/s, respectively, for the top row and $v_{\min} = 30$m/s and $v_{\max} = 40$m/s, respectively, for the bottom row, and number of AVs are 20 and 50 in the top and bottom row, respectively

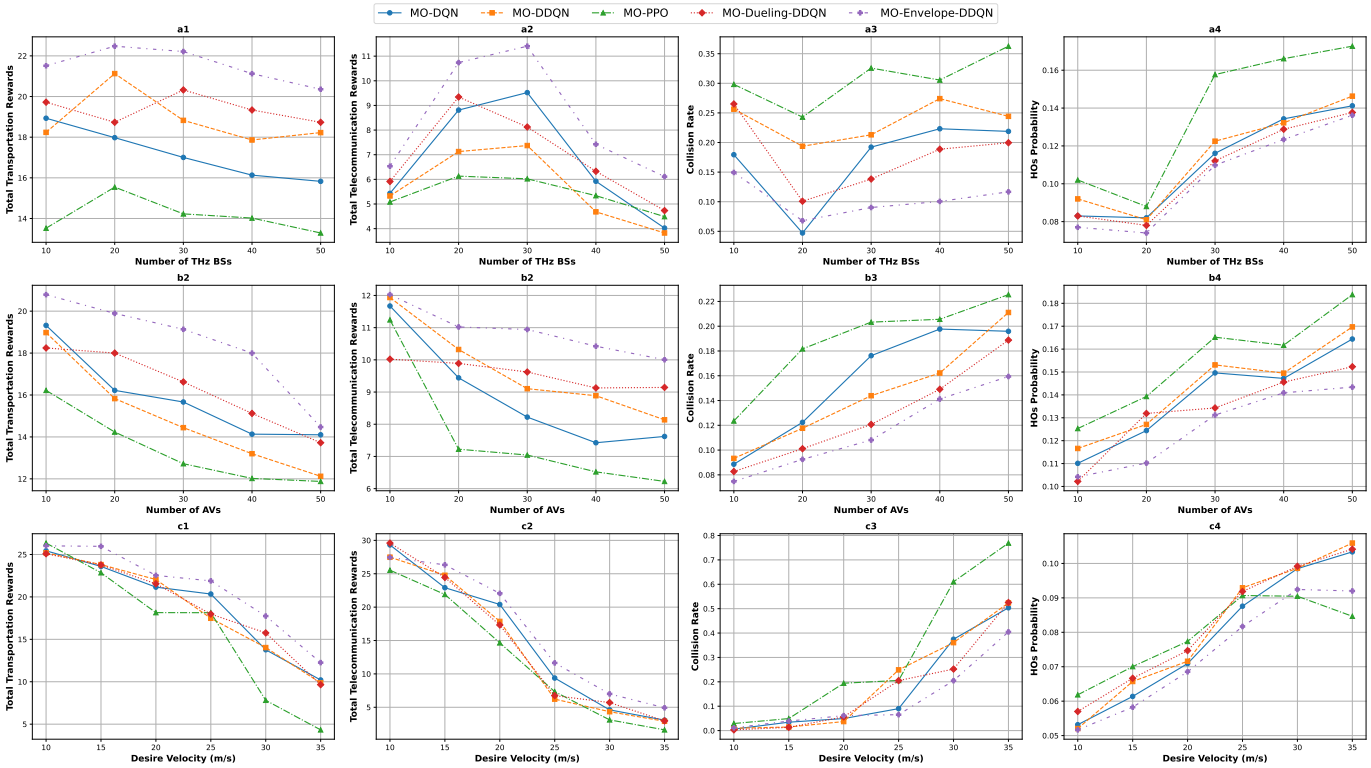


Figure 6: Evaluation performance on (1) total transportation rewards, (2) total telecommunication rewards, (3) collision rate, and (4) HOs probability, as a function of (a) variation in TBSs with counts ranging from 10 to 50 while maintaining an AV velocity of 25 m/s for 20 AVs, and (b) variation in vehicle numbers adjusting from 10 to 50 with a fixed AV velocity of 25 m/s alongside 20 TBSs. (c) variation in desired velocity from 5 to 30 m/s with fixed 20 AVs and 20 TBSs.

Target Velocity (m/s)	Model	Number of AVs M			
		M=10	M=20	M=30	M=40
15	MO-DQN	23.42/10.21/33.63	22.71/9.18/31.89	21.09/7.45/28.54	18.35/6.12/24.47
	MO-DDQN	24.28/12.31/36.59	**24.89/11.41/36.30**	21.85/8.63/30.48	19.41/7.11/26.52
	MO-DDQN-Envelope	**25.76/13.45/39.21**	23.74/10.03/33.77	**23.16/9.86/33.02**	**20.83/7.89/28.72**
20	MO-DQN	20.85/8.56/29.41	19.92/6.74/26.66	19.23/6.35/25.58	17.48/4.39/21.87
	MO-DDQN	21.62/10.19/31.81	21.04/8.57/29.61	20.11/7.93/28.04	18.72/5.97/24.69
	MO-DDQN-Envelope	**22.49/11.45/33.94**	**21.98/9.92/31.90**	**20.87/8.72/29.59**	**19.61/6.18/25.79**
25	MO-DQN	18.21/7.48/25.69	17.45/5.91/23.36	16.34/5.12/21.46	14.87/4.37/19.24
	MO-DDQN	18.92/8.19/27.11	17.88/6.72/24.60	16.97/5.94/22.91	15.32/4.92/20.24
	MO-DDQN-Envelope	**19.64/9.01/28.65**	**18.37/7.51/25.88**	**17.04/6.52/23.56**	**16.01/5.38/21.39**
30	MO-DQN	16.09/6.32/22.41	15.76/5.03/20.79	14.92/4.76/19.68	13.81/4.21/18.02
	MO-DDQN	16.72/6.89/23.61	16.21/5.32/21.53	15.43/5.14/20.57	**14.89/4.91/19.80**
	MO-DDQN-Envelope	**17.35/7.45/24.80**	**16.88/5.97/22.85**	**15.97/5.63/21.60**	14.12/4.49/18.61

Table IV: Evaluation performance of MO-DQN, MO-DDQN, and MO-DDQN-Envelope (total transportation reward / total telecommunication reward / total reward).

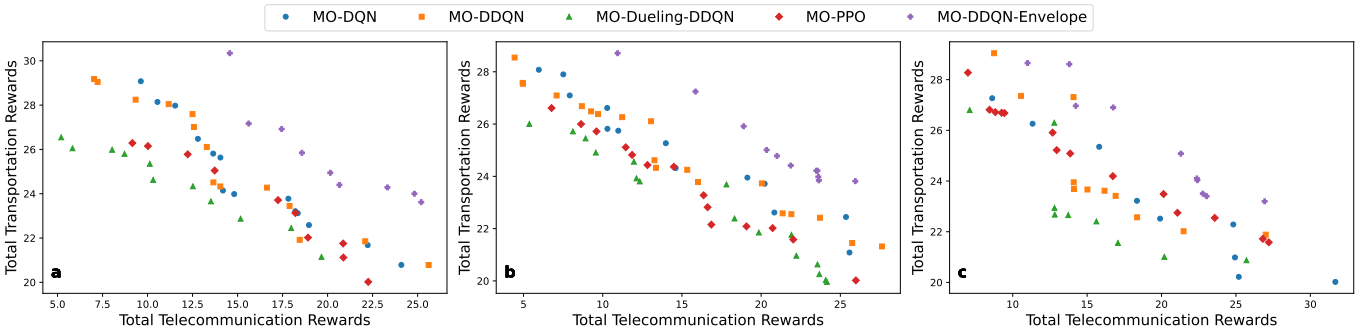


Figure 7: Pareto Frontier Comparison in MOO for total Transportation reward (R^{tran}) and total telecommunication reward (R^{tele}) among MO-DQN, MO-DDQN, MO-dueling-DDQN, MO-PPO, and MO-DDQN-Envelop, across instances:(a) I-(20,30,20,20), (b) I-(20,30,10,20), (c) I-(20,30,20,50)

telecommunication reward has no impact on the training process.

- **Transportation Only:** This scenario focuses solely on the transportation reward, with the telecommunication reward not influencing the training.
- **Telecommunication+Transportation:** This scenario incorporates both the transportation and telecommunication rewards jointly during the training process.

It is important to note that the time consumption of Telecommunication+Transportation is reasonable and comparable to the system policy update 0.067 s (15 Hz) [24]. In [24], [34], [42], the authors considered AV motion updates at a rate of 15 Hz, i.e., 0.067s or 67 ms. Note that, the update frequency is not dependent on the desired velocity. On the other hand, according to uRLLC (5), the transmission latency is given by $D_t = \frac{L_B}{W}$, where $W = W_T$ for TBSs and $W = W_R$ for RBSs, thus $D_t = 5 \times 10^{-8}$s for TBSs and $D_t = 4 \times 10^{-7}$s for RBSs [48], [49], which are relatively smaller than 0.0667s. As such, the telecommunication actions can update at the same time scale as the AV motion action updates. Thus, updating and synchronizing telecommunication and AV motion updates in environments such as *highway-env* [24] or the proposed *RF-THz-highway-env* is reasonable.

7) Simultaneously Multi Objective Optimization Analysis: As depicted in Figure 8, we consider the three scenarios to examine the impact of optimizing different objectives. we found that considering both telecommunication and transportation objectives performs well compared to relying on

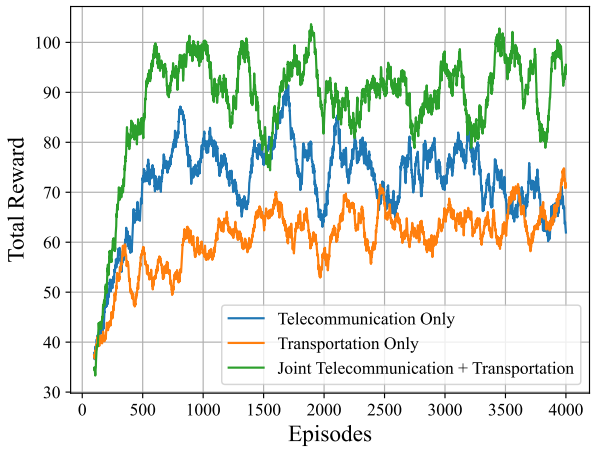


Figure 8: MO-Envelope-DDQN performance comparison with (1) Telecommunication optimization only, (2) Transportation optimization only, and (3) Joint Telecommunication + Transportation optimization.

Instance	Training (s)		Evaluation (s)	
	Transportation	Telecommunication + Transportation	Transportation	Telecommunication + Transportation
I-(20,30,20,20)	0.0743	0.0754	0.0643	0.0654
I-(25,35,20,20)	0.0724	0.0731	0.0614	0.0613
I-(20,30,10,20)	0.0764	0.0784	0.0637	0.0648
I-(20,30,20,50)	0.0826	0.0845	0.0667	0.0672
I-(30,40,20,20)	0.0858	0.0903	0.0725	0.0736

Table V: Comparison of Training and Evaluation Times for Different Instances (in seconds).

Telecommunication Only and Transportation Only alone.

8) MORL Benchmarks Training Time Comparison: We trained each algorithm for 4000 episodes on the test instances given in Table III and recorded the total training time, as summarized in Table VI. Our proposed MO-Envelope-DDQN demonstrates competitive training times while offering significant improvements in multi-objective optimization performance. Our results in Table VI show that the training time is mostly comparable to the baselines, if not significantly less.

VI. Conclusion

We introduce a novel MORL framework tailored for devising joint network selection and autonomous driving policies within a multi-band VNet. Our goals encompass enhancing traffic flow, minimizing collisions, maximizing data rates, and minimizing handoffs (HOs). We achieve this through controlling vehicle motion dynamics and network selection, employing a unique reward function that optimizes data rate, traffic flow, load balancing, and penalizes HOs and unsafe driving behaviors. The problem is formalized as a MOMDP, integrating telecommunication and autonomous driving utilities in its rewards. We propose single policy and multi-policy MORL solutions with predefined and unknown preferences. Numerical results demonstrate the superiority of our proposed solution over weighted sum-based MORL solutions with DQN, showcasing improvements of 12.7%, 18.9%, and 12.3% on average transportation reward, average telecommunication reward, and average HO rate, respectively. Future research could enhance the generalization of the MORL model to better adapt to dynamic traffic conditions using well-established strategies such as meta-learning [50].

Acknowledgment

The authors would like to thank Dr. Hongda Wu for his valuable insights during the initial discussions and his contributions to this work's development and improvement.

Appendix

Given a set of preferences in MORL problems, single policy algorithms aim to scalarize the reward value to determine the best policy, considering the relative priorities assigned to competing objectives. We explore two DRL methods: DQN and DDQN for MORL, each of which employs a neural network parameterized by $\boldsymbol{\theta}_t$ (in each time step t) to approximate the Q-value function for a state-action pair, i.e., $Q(s_t, a_t; \boldsymbol{\theta}_t)$. After taking action a_t in state s_t and receiving instant reward r_{t+1}, we can formulate a target Q function,

$$\hat{Q}(s_t, a_t) = \mathbb{E}_\pi \left[r_t(s_t, a_t) + \gamma Q_\pi(s_{t-1}, a_{t-1}) \right] \tag{26}$$

which is used to optimize the neural network $\boldsymbol{\theta}_t$ using gradient descent, as given in [51],

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + \kappa \left(\hat{Q}(s_t, a_t) - Q(s_t, a_t; \boldsymbol{\theta}_t) \right) \nabla_{\boldsymbol{\theta}_t} Q(s_t, a_t; \boldsymbol{\theta}_t), \tag{27}$$

where κ is a positive scalar representing the learning rate. To learn a single policy for multiple tasks, we scalarize the reward vector by applying the predefined priority of each objective function [1], where a weighted reward function $r_t = \sum_{j=1}^{M_1} r_t^{\mathrm{j,tran}} + \sum_{j=1}^{M_1} r_t^{\mathrm{j,tele}}$ is defined to facilitate the conversion of multi-dimensional rewards into a scalar value.

The MO-DQN method incorporates a target Q-network and experience replay to stabilize the learning process and ensure convergence, as discussed in the following:

- *Target Network:* Another set of neural network $\boldsymbol{\theta}_t^-$ is introduced to compute target Q value at each time step t, which has the same architecture as $\boldsymbol{\theta}_t$, but with frozen parameters. Specifically, $\boldsymbol{\theta}_t^-$ only copies those parameters from $\boldsymbol{\theta}_t$ every N^- steps and remains fixed until the next scheduled update [52]. The target value for the MO-DQN is defined as:

$$\hat{Q}(s_t, a_t) = r_{t+1} + \gamma \arg\max_a Q(s_{t+1}, a; \boldsymbol{\theta}_t^-) \tag{28}$$

- *Experience Replay:* To address issues related to correlations between sequential observations and to improve data efficiency, MO-DQN utilizes the experience replay mechanism, which stores past transition tuples (s_z, a_z, s_{z+1}, r_z) in a replay buffer $\mathcal{D}_\mathcal{T}$ with size $N_\mathcal{T}$, i.e., $z \in \{1, \ldots, N_\mathcal{T}\}$. During the training phase, mini-batches of these transitions are randomly sampled from the buffer. This method not only reduces the variance of each update but also allows the neural network to benefit from learning across a diverse range of past experiences, thus avoiding local optima and overfitting.

Unlike MO-DQN, where the current weights $\boldsymbol{\theta}_t$ are used both to select and evaluate actions, MO-DDQN utilizes a separate set of parameter $\boldsymbol{\theta}_t'$ to evaluate the value of the policy, ensuring a more reliable estimate by decoupling the selection and evaluation of actions. Given $\boldsymbol{\theta}_t$ and $\boldsymbol{\theta}_t'$ corresponding to evaluation and target Q networks, respectively, the target value function in MO-DDQN [51] is updated as follows:

$$\hat{Q}(s_t, a_t) = r_t + \gamma Q \left(s_{t+1}, \arg\max_a Q(s_{t+1}, a; \boldsymbol{\theta}_t); \boldsymbol{\theta}_t' \right) \tag{29}$$

where the action selection is guided by the online weights $\boldsymbol{\theta}_t$.

In MO-DQN and MO-DDQN, the neural network parameterized by $\boldsymbol{\theta}_t$ associated with evaluation function is updated

Instance	MO-Q [45]	MO-DQN [1]	MO-Dueling-DDQN [12]	MO-PPO [46]	EUPG [47]	MO-Envelope-DDQN
I-(20,30,20,20)	30:36	31:10	30:45	31:50	32:05	**30:15**
I-(25,35,20,20)	27:24	27:22	28:59	28:41	28:17	**26:25**
I-(20,30,10,20)	30:11	30:53	30:32	31:22	31:57	**30:06**
I-(20,30,20,50)	32:00	32:20	31:50	33:00	33:15	**31:45**
I-(30,40,20,20)	21:45	22:08	21:33	22:52	23:35	**21:17**

Table VI: Comparison of Training Times for Different MORL Methods. The times reported in the tables are formatted as **MM:SS**, where **MM** represents the number of minutes and **SS** represents the number of seconds.

by minimizing the mean square error loss $\mathcal{L}(\boldsymbol{\theta})$ between Q and $\hat{Q}$ as follows [52]:

$$\mathcal{L}(\boldsymbol{\theta}_t) = \mathbb{E}_{z\in\mathcal{D}_\mathcal{T}}\left(Q(s_z, a_z; \boldsymbol{\theta}_t) - \hat{Q}(s_z, a_z)\right)^2 \tag{30}$$

The proposed MO-DQN and MO-DDQN are illustrated in Fig. 3. The training algorithm of the proposed MO-DQN and MO-DDQN are in **Algorithm** 2.

REFERENCES

[1] Z. Yan and H. Tabassum, "Reinforcement learning for joint V2I network selection and autonomous driving policies," in *Proc. 2022 IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2022, pp. 1241–1246.

[2] H. Ye, G. Y. Li, and B.-H. F. Juang, "Deep reinforcement learning based resource allocation for V2V communications," *IEEE Trans. Veh. Technol.*, vol. 68, no. 4, pp. 3163–3173, Feb. 2019.

[3] L. Liang, H. Ye, and G. Y. Li, "Spectrum sharing in vehicular networks based on multi-agent reinforcement learning," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 10, pp. 2282–2292, Aug. 2019.

[4] Y. Xu, K. Zhu, H. Xu, and J. Ji, "Deep reinforcement learning for multi-objective resource allocation in multi-platoon cooperative vehicular networks," *IEEE Trans. Wireless Commun.*, Feb. 2023.

[5] Z. Yan, H. Zhou, H. Tabassum, and X. Liu, "Hybrid LLM-DDQN-based joint optimization of V2I communication and autonomous driving," *IEEE Wireless Commun. Lett.*, vol. 14, no. 4, pp. 1214–1218, Feb. 2025.

[6] X. Hu, Y. Zhang, X. Liao, Z. Liu, W. Wang, and F. M. Ghannouchi, "Dynamic beam hopping method based on multi-objective deep reinforcement learning for next generation satellite broadband systems," *IEEE Trans. on Broadcasting*, vol. 66, no. 3, pp. 630–646, Jan. 2020.

[7] G. Yu, Y. Jiang, L. Xu, and G. Y. Li, "Multi-objective energy-efficient resource allocation for multi-rat heterogeneous networks," *IEEE J. Sel. Areas Commun.*, vol. 33, no. 10, pp. 2118–2127, May 2015.

[8] R. Devarajan, S. C. Jha, U. Phuyal, and V. K. Bhargava, "Energy-aware resource allocation for cooperative cellular network using multi-objective optimization approach," *IEEE Trans. Wireless Commun.*, vol. 11, no. 5, pp. 1797–1807, Mar. 2012.

[9] 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 Trans. Veh. Technol.*, vol. 69, no. 11, pp. 13 124–13 138, Sep. 2020.

[10] H. Khan, A. Elgabli, S. Samarakoon, M. Bennis, and C. S. Hong, "Reinforcement learning-based vehicle-cell association algorithm for highly mobile millimeter wave communication," *IEEE Trans. on Cogn. Commun. Netw.*, vol. 5, no. 4, pp. 1073–1085, Sep. 2019.

[11] Z. Wu, K. Qiu, and H. Gao, "Driving policies of V2X autonomous vehicles based on reinforcement learning methods," *IET Intell. Transp. Syst.*, vol. 14, no. 5, pp. 331–337, Feb. 2020.

[12] Z. Yan, W. Jaafar, B. Selim, and H. Tabassum, "Multi-uav speed control with collision avoidance and handover-aware cell association: Drl with action branching," in *Proc. 2023 IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2023, pp. 5067–5072.

[13] X. Liu, Y. Liu, Y. Chen, and L. Hanzo, "Enhancing the fuel-economy of V2I-assisted autonomous driving: A reinforcement learning approach," *IEEE Trans. Veh. Technol.*, vol. 69, no. 8, pp. 8329–8342, May 2020.

[14] A. Alizadeh, M. Moghadam, Y. Bicer, N. K. Ure, U. Yavas, and C. Kurtulus, "Automated lane change decision making using deep reinforcement learning in dynamic and uncertain highway environment," in *Proc. 2019 IEEE Intell. Transp. Syst. Conf. (ITSC)*. IEEE, Oct. 2019, pp. 1399–1404.

[15] X. He and C. Lv, "Towards energy-efficient autonomous driving: A multi-objective reinforcement learning approach," *IEEE/CAA J. Autom. Sinica*, vol. 10, no. 5, pp. 1329–1331, May 2023.

[16] H. Shoaib, M. Nourinejad, and H. Tabassum, "Macroscopic traffic flow analysis and optimization with V2I connectivity and collision avoidance constraints," in *ICC 2023-IEEE International Conf. on Communications*. IEEE, May 2023, pp. 323–328.

[17] H. Shoaib and H. Tabassum, "Optimization of speed and network deployment for reliable V2I communication in the presence of handoffs and interference," *IEEE Wireless Commun. Lett.*, vol. 12, no. 6, pp. 1051–1055, 2023.

[18] M. A. Saeidi, H. Shoaib, and H. Tabassum, "A tractable handoff-aware rate outage approximation with applications to thz-enabled vehicular network optimization," in *Proc. 2023 IEEE Global Commun. Conf. (GLOBECOM)*. IEEE, Mar. 2023, pp. 5092–5097.

[19] J. Pei, J. Wang, D. Shi, and P. Wang, "Detection and imputation-based two-stage denoising diffusion power system measurement recovery under cyber-physical uncertainties," *IEEE Trans. Smart Grid*, vol. 15, no. 6, pp. 5965–5980, May 2024.

[20] J. Pei, C. Feng, P. Wang, H. Tabassum, and D. Shi, "Latent diffusion model-enabled low-latency semantic communication in the presence of semantic ambiguities and wireless channel noises," *IEEE Trans. Wireless Commun.*, vol. 24, no. 5, pp. 4055–4072, Feb. 2025.

[21] F. Song, H. Xing, X. Wang, S. Luo, P. Dai, Z. Xiao, and B. Zhao, "Evolutionary multi-objective reinforcement learning based trajectory control and task offloading in uav-assisted mobile edge computing," *IEEE Trans. Mobile Comput.*, Sep. 2022.

[22] C. Li and K. Czarnecki, "Urban driving with multi-objective deep reinforcement learning," *arXiv preprint arXiv:1811.08586*, Nov. 2018.

[23] W. Wei, R. Yang, H. Gu, W. Zhao, C. Chen, and S. Wan, "Multi-objective optimization for resource allocation in vehicular cloud computing networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 12, pp. 25 536–25 545, Aug. 2021.

[24] E. Leurent, "An environment for autonomous driving decision-making," https://github.com/eleurent/highway-env, 2018.

[25] P. G. Gipps, "A behavioural car-following model for computer simulation," *Transp. Res. Part B Methodol.*, vol. 15, no. 2, pp. 105–111, Apr. 1981. [Online]. Available: http://dx.doi.org/10.1016/0191-2615(81)90037-0

[26] Y. Zhang, X. Chen, J. Wang, Z. Zheng, and K. Wu, "A generative car-following model conditioned on driving styles," *Transportation Research Part C: Emerging Technologies*, vol. 145, p. 103926, 12 2022. [Online]. Available: http://dx.doi.org/10.1016/j.trc.2022.103926

[27] H. Wei and H. Zhang, "An equivalent model for handover probability analysis of irs-aided networks," *IEEE Trans. Veh. Technol.*, vol. 72, no. 10, pp. 13 770–13 784, May 2023.

[28] M. T. Hossan and H. Tabassum, "Mobility-aware performance in hybrid rf and terahertz wireless networks," *IEEE Trans. Commun.*, vol. 70, no. 2, pp. 1376–1390, Dec. 2021.

[29] M. A. Saeidi, H. Tabassum, and M. Alizadeh, "Molecular absorption-aware user assignment, spectrum, and power allocation in dense thz networks with multi-connectivity," *IEEE Trans. Wireless Commun.*, Aug. 2024.

[30] C. She, C. Sun, Z. Gu, Y. Li, C. Yang, H. V. Poor, and B. Vucetic, "A tutorial on ultrareliable and low-latency communications in 6g: Integrating domain knowledge into deep learning," *Proc. IEEE*, vol. 109, no. 3, pp. 204–246, Mar. 2021.

[31] M. Monemi and H. Tabassum, "Performance of uav-assisted d2d networks in the finite block-length regime," *IEEE Trans. Commun.*, vol. 68, no. 11, pp. 7270–7285, Aug. 2020.

[32] P. Polack, F. Altché, B. d'Andréa Novel, and A. de La Fortelle, "The kinematic bicycle model: A consistent model for planning feasible trajectories for autonomous vehicles?" in *Proc. 2017 IEEE Intell. Veh. Symp. (IV)*. IEEE, Jun. 2017, pp. 812–818.

[33] A. Kesting, M. Treiber, and D. Helbing, "General lane-changing model mobil for car-following models," *Transportation Research Record*, vol. 1999, no. 1, pp. 86–94, Jan. 2007.

Algorithm 2: MO-DDQN Algorithm

Result: Learned action-value function $Q_{\boldsymbol{\theta}}$ and Policy π
Data: Evaluation Q-network Q with weights $\boldsymbol{\theta}$, Target Q-network $\hat{Q}$ with weights $\boldsymbol{\theta}'$ (for MO-DDQN only), Experience replay memory $\mathcal{D}_{\mathcal{T}}$, Mini-batch size $N_{\mathcal{T}}$, Horizon limit of each episode T_{hl}.
Initialization:
Experience replay memory $\mathcal{D}_{\mathcal{T}} \leftarrow \emptyset$,
Initialize Q-network weights $\boldsymbol{\theta}$ randomly,
For MO-DDQN: Initialize target network weights $\boldsymbol{\theta}' \leftarrow \boldsymbol{\theta}$,
Initialize $Q(s, a)$ for all states s and actions a, including AVs, TBSs, and RBSs.
while *episode* $<$ *episode limit and runtime* $<$ *time limit* **do**
  Initialize $t \leftarrow 0$ and state s_t based on environment
  while $t \leq T_{hl}$ **do**
    RL agent select a_t from $\mathcal{A}$ with probability ϵ or select a_t from $\max_{a \in \mathcal{A}} Q(s_t, a_t; \boldsymbol{\theta})$ with probability of $1 - \epsilon$.
    Derive a_t^{tran} and a_t^{tele} from a_t
    Apply a_t^{tran} and a_t^{tele},
    observe reward r_t and next state s_{t+1}.
    Store transition (s_t, a_t, s_{t+1}, r_t) in $\mathcal{D}_{\mathcal{T}}$.
    Experience Replay: Sample a mini-batch of transitions (s_z, a_z, r_z, s_{z+1}) from $\mathcal{D}_{\mathcal{T}}$, where $z \in \{1, \ldots, N_{\mathcal{T}}\}$.
    Set target-Q for each sampled transition:
    for *each transition* z **do**
      if *episode ends at step* $z + 1$ **then**
        $\hat{Q}(s_z, a_z) = r_z$
      else
        Use $\hat{Q}$ to compute $\hat{Q}(s_z, a_z)$ according to MO-DQN or MO-DDQN update by (28), (29).
      end
    end
    Perform a gradient descent step on (30) with respect to network parameters $\boldsymbol{\theta}$
    if *MO-DDQN* **then**
      Update target $\hat{Q}$ weights $\boldsymbol{\theta}' \leftarrow \boldsymbol{\theta}$ every N^- steps;
    end
    $t \leftarrow t + 1$
  end
  Update policy π based on learned Q.
end

[34] E. Leurent, "Safe and efficient reinforcement learning for behavioural planning in autonomous driving," Ph.D. dissertation, Université de Lille, Oct. 2020.

[35] R. Yang, X. Sun, and K. Narasimhan, "A generalized algorithm for multi-objective reinforcement learning and policy adaptation," in *Adv. Neural Inf. Process. Syst.* Curran Associates, Inc., Dec. 2019, vol. 32, pp. 14 610–14 621, NeurIPS 2019.

[36] 3rd Generation Partnership Project (3GPP), "Work Item Description: AI/ML Model Transfer in 5GS (WIID: 1020093)," https://portal.3gpp.org/desktopmodules/WorkItem/WorkItemDetails.aspx?workitemId=1020093, Apr. 2024, 3GPP Work Item, Accessed: 2025-04-19.

[37] M. Andrychowicz, F. Wolski, A. Ray, J. Schneider, R. Fong, P. Welinder, B. McGrew, J. Tobin, P. Abbeel, and W. Zaremba, "Hindsight experience replay," *Adv. Neural Inf. Process. Syst.*, vol. 30, Dec. 2017, NeurIPS 2017.

[38] L. T. Watson and R. T. Haftka, "Modern homotopy methods in optimization," *Comput. Methods Appl. Mech. Eng.*, vol. 74, no. 3, pp. 289–305, Sep. 1989.

[39] L. N. Alegre, F. Felten, E.-G. Talbi, G. Danoy, A. Nowé, A. L. C. Bazzan, and B. C. da Silva, "MO-Gym: A library of multi-objective reinforcement learning environments," in *Proc. 34th Benelux Conf. Artif. Intell. (BNAIC/Benelearn)*, Nov. 2022.

[40] E. Leurent, "rl-agents: Implementations of reinforcement learning algorithms," https://github.com/eleurent/rl-agents, 2018.

[41] F. Felten, L. N. Alegre, A. Nowé, A. L. C. Bazzan, E. G. Talbi, G. Danoy, and B. C. d. Silva, "A toolkit for reliable benchmarking and research in multi-objective reinforcement learning," in *Adv. Neural Inf. Process. Syst.*, Dec. 2023, NeurIPS 2023.

[42] Y. Dong, T. Datema, V. Wassenaar, J. Van de Weg, C. T. Kopar, and H. Suleman, "Comprehensive training and evaluation on deep reinforcement learning for automated driving in various simulated driving maneuvers," in *Proc. 2023 IEEE Intell. Transp. Syst. Conf. (ITSC)*. IEEE, Sep. 2023, pp. 6165–6170.

[43] Z. Wang, T. Schaul, M. Hessel, H. Van Hasselt, M. Lanctot, and N. De Freitas, "Dueling network architectures for deep reinforcement learning," in *Proc. Int. Conf. Mach. Learn. (ICML)*. PMLR, Jun. 2016, pp. 1995–2003, ICML 2016.

[44] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 7 2017.

[45] K. Van Moffaert, M. M. Drugan, and A. Nowé, "Scalarized multi-objective reinforcement learning: Novel design techniques," in *Proc. 2013 IEEE Symp. Adapt. Dyn. Program. Reinforcement Learn. (ADPRL)*. IEEE, Apr. 2013, pp. 191–199.

[46] N. D. H. Khoi, C. P. Van, H. V. Tran, and C. D. Truong, "Multi-objective exploration for proximal policy optimization," in *Proc. 2020 Appl. New Technol. Green Build. (ATiGB)*. IEEE, May 2021, pp. 105–109.

[47] D. M. Roijers, D. Steckelmacher, and A. Nowé, "Multi-objective reinforcement learning for the expected utility of the return," in *Proc. Adapt. Learn. Agents Workshop at FAIM*, vol. 2018, 7 2018, FAIM Workshop 2018.

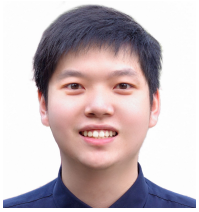
[48] A. Lancho, J. Östman, G. Durisi, T. Koch, and G. Vazquez-Vilar, "Saddlepoint approximations for short-packet wireless communications," *IEEE Trans. Wireless Commun.*, vol. 19, no. 7, pp. 4831–4846, Apr. 2020.

[49] M. Shirvanimoghaddam, M. S. Mohammadi, R. Abbas, A. Minja, C. Yue, B. Matuz, G. Han, Z. Lin, W. Liu, Y. Li *et al.*, "Short block-length codes for ultra-reliable low latency communications," *IEEE Commun. Mag.*, vol. 57, no. 2, pp. 130–137, Dec. 2018.

[50] A. A. Al-Habob, H. Tabassum, and O. Waqar, "Non-orthogonal age-optimal information dissemination in vehicular networks: A meta multi-objective reinforcement learning approach," *IEEE Trans. on Mobile Comput.*, vol. 23, no. 10, pp. 9789–9803, Oct. 2024.

[51] H. Van Hasselt, A. Guez, and D. Silver, "Deep reinforcement learning with double q-learning," in *Proc. AAAI Conf. Artif. Intell.*, vol. 30, no. 1, Feb. 2016, AAAI 2016.

[52] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski *et al.*, "Human-level control through deep reinforcement learning," *nature*, vol. 518, no. 7540, pp. 529–533, Feb. 2015.



Zijiang Yan (Graduate Student Member, IEEE) received the B.S. degree with a double major in Computer Science and Statistics from York University, Toronto, ON, Canada, in 2021. He is currently a Research Assistant in the Department of Electrical Engineering and Computer Science, Lassonde School of Engineering, York University. His research interests include AI-enabled communications, Quantum Machine Learning, Diffusion Models, and Large Language Models. He received third place in the 2025 Student Innovation Competition on Sustainable Space Communications, hosted by the Satellite and Space Communications Technical Committee (SSC TC) of the IEEE Communications Society (IEEE COMSOC). He also received the Lassonde Undergraduate Research Award (LURA) from York University in 2021. In addition, he has served as a session chair at multiple flagship conferences, such as IEEE GLOBECOM 2023 and IEEE ICC 2025.

Hina Tabassum (Senior Member, IEEE) (M'12–SM'18) received the Ph.D. degree from the King Abdullah University of Science and Technology (KAUST). She is currently an Associate Professor with the Lassonde School of Engineering, York University, Canada, where she joined as an Assistant Professor in 2018. Prior to that, she was a Postdoctoral Research Associate at the University of Manitoba, Canada. She is also appointed as a Visiting Faculty at the University of Toronto in 2024 and the York Research Chair of 5G/6G-enabled mobility and sensing applications in 2023, for five years. She is also appointed as the IEEE COMSOC Distinguished Lecturer for the term 2025–2026. She is listed in Stanford's list of the World's Top Two-Percent Researchers (2021–2024). She received the Lassonde Innovation Early-Career Researcher Award in 2023 and was recognized by N2WOMEN as a Rising Star in Computer Networking and Communications in 2022. She has published over 100 refereed articles in well-reputed IEEE journals, magazines, and conferences. She has received multiple Exemplary Editor awards from IEEE COMMUNICATIONS LETTERS (2020), IEEE OPEN JOURNAL OF THE COMMUNICATIONS SOCIETY (OJCOMS) (2023–2024), and IEEE TRANSACTIONS ON GREEN COMMUNICATIONS AND NETWORKING (2023). She was also named an Exemplary Reviewer (Top 2%) for IEEE TRANSACTIONS ON COMMUNICATIONS in 2015, 2016, 2017, 2019, and 2020. She is the Founding Chair of the Special Interest Group on THz Communications in the IEEE COMMUNICATIONS SOCIETY (COMSOC) Radio Communications Committee (RCC). She served as an Associate Editor for IEEE COMMUNICATIONS LETTERS (2019–2023), IEEE OPEN JOURNAL OF THE COMMUNICATIONS SOCIETY (OJCOMS) (2019–2023), and IEEE TRANSACTIONS ON GREEN COMMUNICATIONS AND NETWORKING (2020–2023). She is currently serving as an Area Editor for IEEE OJCOMS, and as an Associate Editor for IEEE TRANSACTIONS ON COMMUNICATIONS, IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, IEEE TRANSACTIONS ON MOBILE COMPUTING, and IEEE COMMUNICATIONS SURVEYS & TUTORIALS.