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

Zijiang Yan, Hina Tabassum

## Contributions

- 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 then propose single policy MORL solutions with predefined preferences thus converting the MOOP into a single-objective and apply DQN and double DQN solutions. The resulting optimal policy depends on the relative preferences of the objectives.

- Learning optimized policies across **multiple preferences** remains challenging. To address this, we then develop a novel envelope MORL solution to effectively navigate the entire spectrum of preferences within a given domain. This approach empowers the trained model to generate the best possible policy tailored to any user-defined preference. Our algorithm hinges on two fundamental insights: firstly, we demonstrate that the **optimality operator** governing a generalized Bellman equation with preferences exhibits valid contraction properties. Secondly, by optimizing for the **convex envelope of multi-objective Q-values**, we ensure an efficient alignment between preferences and the resultant optimal policies. **Leveraging hindsight experience replay**, we recycle transitions to facilitate learning across various sampled preferences, while employing homotopy optimization to maintain manageable learning processes.

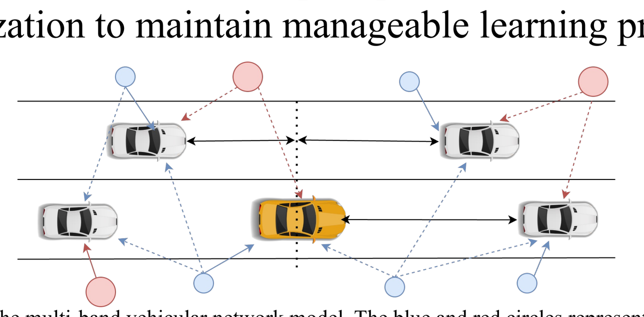

Figure 1: An illustrative structure of the multi-band vehicular network model. The blue and red circles represent TBSs and RBSs, respectively. The solid and dash line represent desired signal links and interference links, respectively.

## System Model and Assumption

- **Kinematics Model:** $\frac{\partial}{\partial t}(x_j) = v_j \cos(\psi_j + \beta_j), \quad \beta_j = \arctan\left(\frac{\tan \delta_j^{\mathrm{fa}}}{2}\right)$

$$\frac{\partial}{\partial t}(y_j) = v_j \sin(\psi_j + \beta_j)$$

$$\frac{\partial}{\partial t}(v_j) = a_j, \quad \frac{\partial}{\partial t}(\psi_j) = \frac{v_j}{l_j} \sin \beta_j$$

- **Acceleration and Lane Change**

$$\frac{\partial}{\partial t}(\psi_j) = K_j^{\psi}\left[\left(\psi_{L_j} + \arcsin\left(\frac{\tilde{v}_{i,y}}{v_j}\right) - \psi_j\right]$$

$$a_j = K_0^{v}(v_r - v_j)$$

- **Network Composition:** two-tier downlink network with $N_R$ RF BSs (RBSs) and $N_T$ THz BSs (TBSs) supporting $V$ (AVs) on a four-lane highway.

- **Bandwidth and Data Rate:** Each BS, whether RBS or TBS, is allocated a specific bandwidth ($W_R$ or $W_T$ ), and data rates are computed as

$$R_{ij} = \frac{W_j}{\ln 2}\left[\ln(1 + \mathrm{SINR}_{ij}) - \sqrt{\frac{V}{L_B}} f_Q^{-1}(\epsilon_c)\right] \qquad \mathrm{WR}_{ij} = \frac{R_{ij}}{\min(Q_i, n_i)}(1-\mu)$$

- **BS Quota and Selection:** Maximum AV limits for each RBS and TBS are denoted by $Q_R$ and $Q_T$ respectively. Each AV maintains a set of top three BSs based on data rates, provided $\mathrm{SINR}_{ij}(t) \geq \gamma_{th}$

- **Handoff Management:** AVs may switch BSs based on SINR requirements impacting data rates due to handoff (HO) latencies. A HO penalty $\mu$ is imposed to discourage frequent HOs, higher for TBSs and lower for RBSs.

## MOMDP Formulation

- **State Space:** position, velocity, number of AVs associated with BS $i$, and their respective SINRs with BSs.

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

- **2D Action Space:** lane changes, acceleration, stop, and deceleration. Communication Action includes different strategies for selecting BS.

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

- **Reward Functions:**

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

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

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

where $\delta_2$ is collision factor , $\xi_t^j$ is HO probability

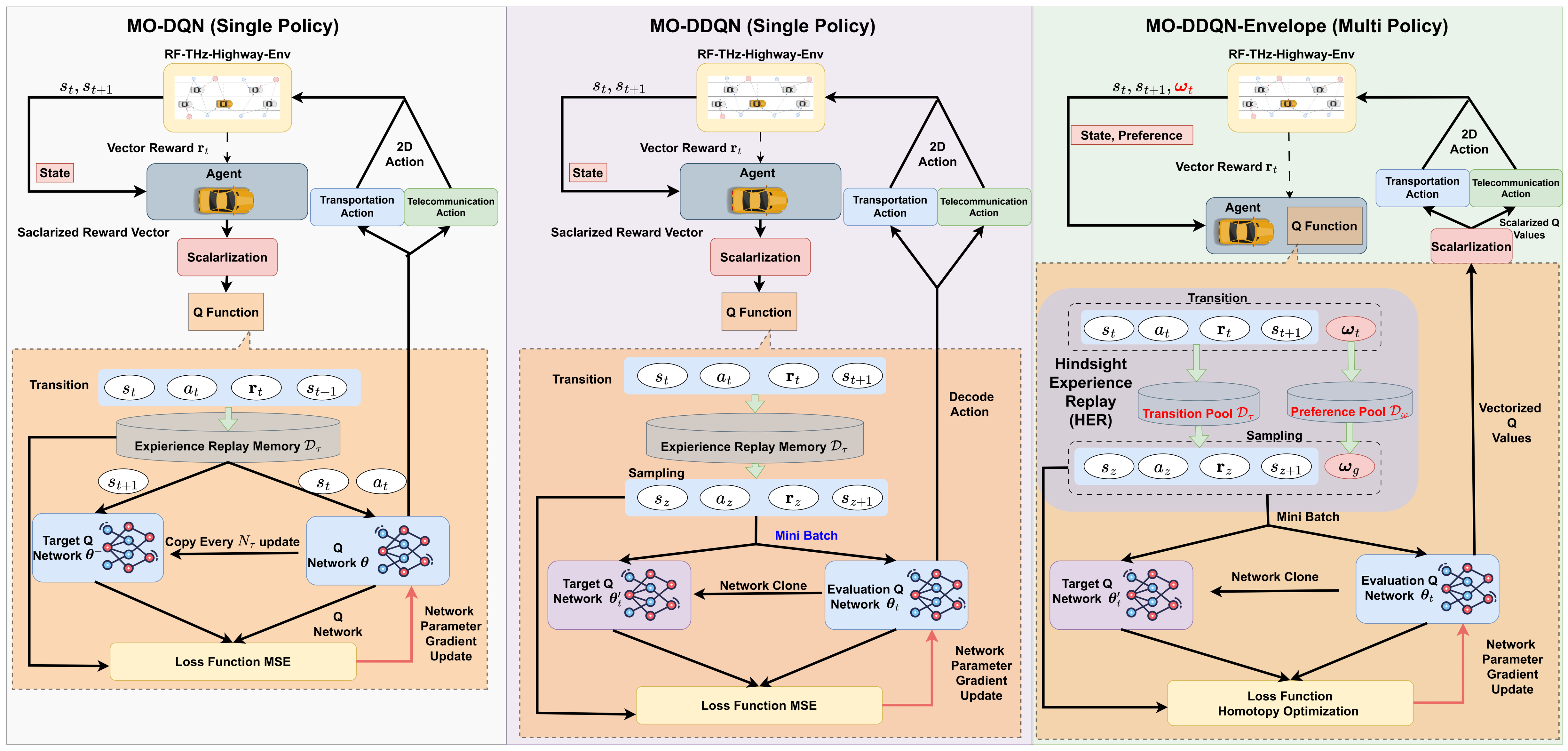


**Figure 2:** Comparison of MO-DQN, MO-DDQN, and the proposed MO-DDQN-envelope framework

## Proposed MO-DDQN-Envelope Solution

**Algorithm 1** Multi-Objective Double Deep Q-Learning

**Result:** Learned action-value function $Q_{\boldsymbol{\theta}}$ and Policy $\pi$
**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 $\epsilon$ or select $a_t$ from $\max_{a \in \mathcal{A}} Q(s_t, a_t; \boldsymbol{\theta})$ with probability of $1-\epsilon$.
    Derive $a_t^{\mathrm{tran}}$ and $a_t^{\mathrm{tele}}$ from $a_t$
    Apply $a_t^{\mathrm{tran}}$ and $a_t^{\mathrm{tele}}$,
    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 $z \in \{1, \dots, N_{\mathcal{T}}\}$.
    **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 (24), (25).
      **end**
    **end**
    Perform a gradient descent step on (26) 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 $\pi$ based on learned $Q$.
**end**

**Algorithm 2** Multi-Objective Envelope DDQN

**Result:** Learned action-value function $\mathbf{Q}_\theta$ and Policy $\pi$
**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 $p_\lambda$
**Initialization:**
HER replay buffer $\mathcal{D}_{\mathcal{T}} \leftarrow \emptyset$,
Initialize Q-network weights $\theta$ 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 $\epsilon$ 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 $a_t^{\mathrm{tran}}$ and $a_t^{\mathrm{tele}}$ from $a_t$;
      Apply $a_t^{\mathrm{tran}}$ and $a_t^{\mathrm{tele}}$ 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 $N_\omega$ 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:
      $\begin{cases} \mathbf{r}_z, & \text{if } s_{z+1} \text{ is terminal} \\ (28), & \text{otherwise} \end{cases}$
      $\forall z \in [1, N_{\mathcal{T}}]$ and $\forall g \in [1, N_\omega]$
      **Homotopy Optimization:**
      Update $\mathbf{Q}_\theta$ by minimizing the loss with gradient descent by (32);
      Gradually increase $\lambda$ following the path $p_\lambda$;
    **end**
    Update target $\mathbf{Q}_{\theta'}$ weights $\theta' \leftarrow \theta$ by (33) every $N^-$ steps;
  **end**
  $t \leftarrow t + 1$
  Compute policy $\pi$ based on learned $\mathbf{Q}_\theta$;
**end**

- **Bellman Operation with Optimal Filter:** The MO optimality operator , as given by:

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

$$(\mathcal{H}\mathbf{Q})_\pi(s, a, \boldsymbol{\omega}) = \arg_Q \sup_{a \in \mathcal{A}, \boldsymbol{\omega}' \in \Omega} \mathbf{Q}_\pi(s, a, \boldsymbol{\omega}').$$

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

Where The optimal filter $H$ is instrumental in solving the convex envelope of PPF, which represents the current solution frontier. This process is key in optimizing the Q-function, $Q_\pi$ for a given state s and preference weights $\omega$.

- **Hindsight Experience Replay :** Transitions and preferences sampling:

$$(s_z, a_z, \mathbf{r}_z, s_{z+1}) \sim \mathcal{D}_{\mathcal{T}}, \text{ where } z \in [1, N_{\mathcal{T}}].$$

$$\mathcal{W} \equiv \{\omega_g \sim \mathcal{D}_\omega\}, \text{ with } g \in [1, N_\omega]$$

Where $D_{\mathrm{T}}$ is the experience replay transition pool and $D_\omega$ is the preference pool

- **Homotopy Optimization :** The MO-DDQN-Envelope , is defined by:

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

The loss function $L^A(\theta)$ focus on the accuracy and correctness of training

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

The loss function $L^B(\theta)$ focus on the smoothness of training

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

To focus on accuracy in the initial training and focus on smoothness afterwards.

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

And the parameters update as

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

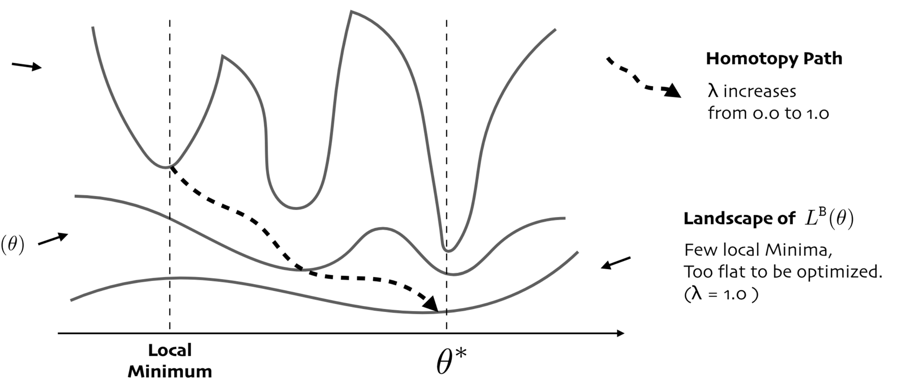


Figure 3: An explanation for homotopy optimization method used in the envelope deep MORL algorithm. The MSE loss $L^A(\theta)$ is hard for optimization since there are many local minima over its landscape. Although the value metric loss $L^B(\theta)$ has fewer local minima, it is also hard for optimization since there are many vectors Q minimizing value metric d. The landscape iof $L^B(\theta)$ is too flat. The homotopy path connecting $L^A(\theta)$ and $L^B(\theta)$ provides better opportunities to find the global optimal parameters $\theta^*$

Paper Website Demos

## Simulation Results and Evaluation

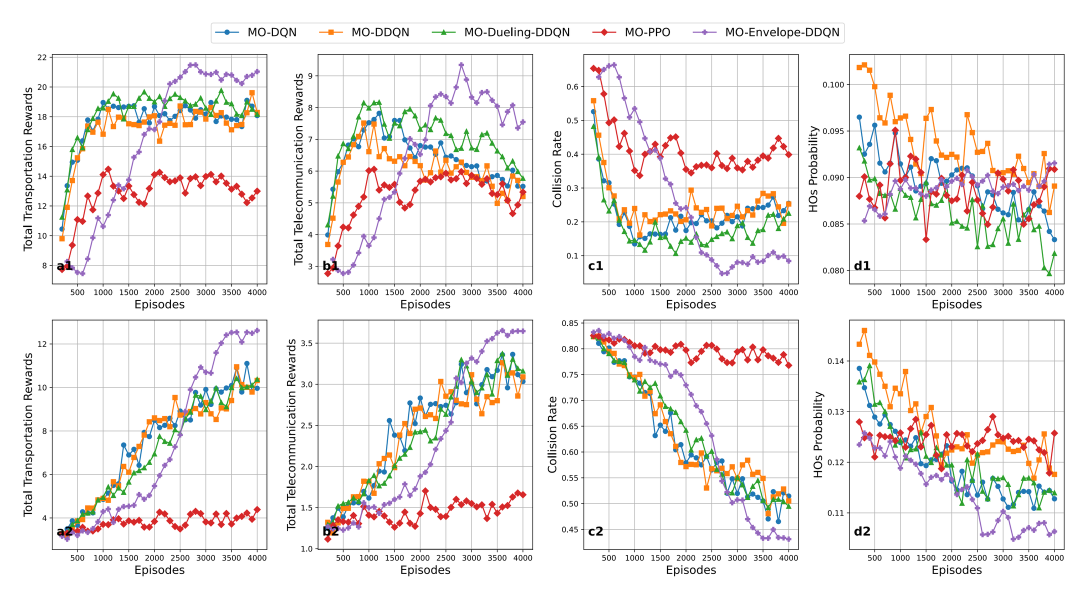


Figure 4: Training performance on (a) total transportation rewards, (b) total telecommunication rewards, (c) collision rate, and (d) HOs probability.

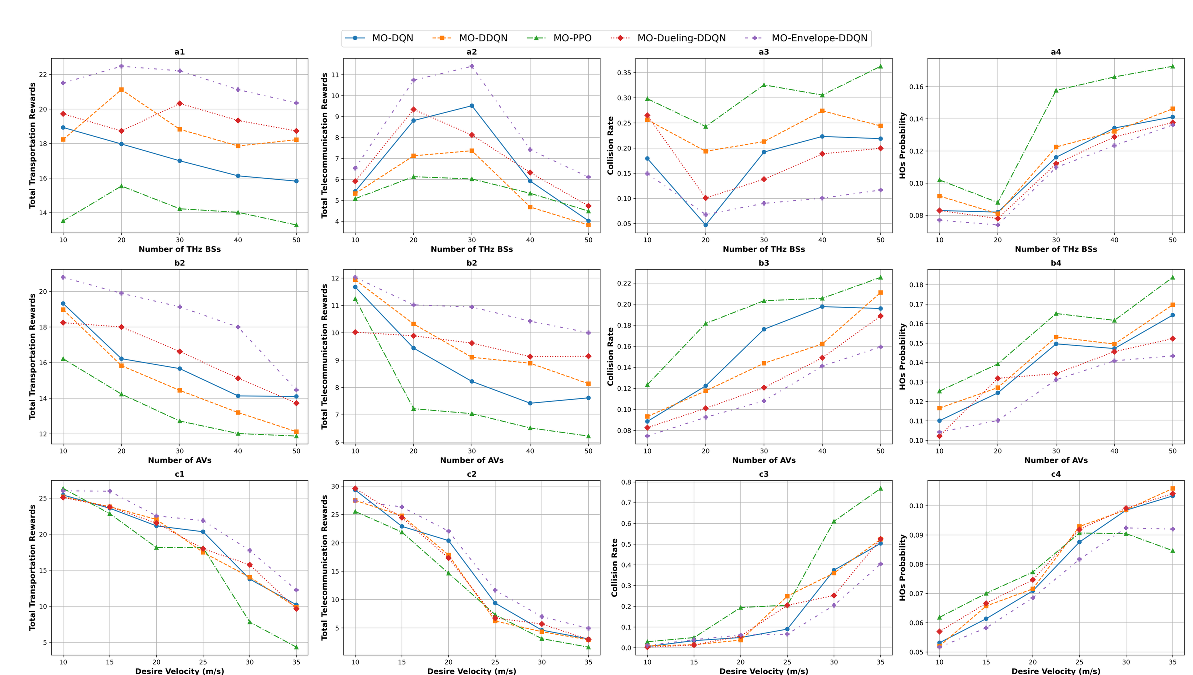


Figure 5: valuation performance on (1) total transportation rewards, (2) total telecommunication rewards, (3) collision rate, and (4) HOs probability, as a function of (a) variation in TBSs , (b) Variation in different number of AVs, (c ) different desire speeds

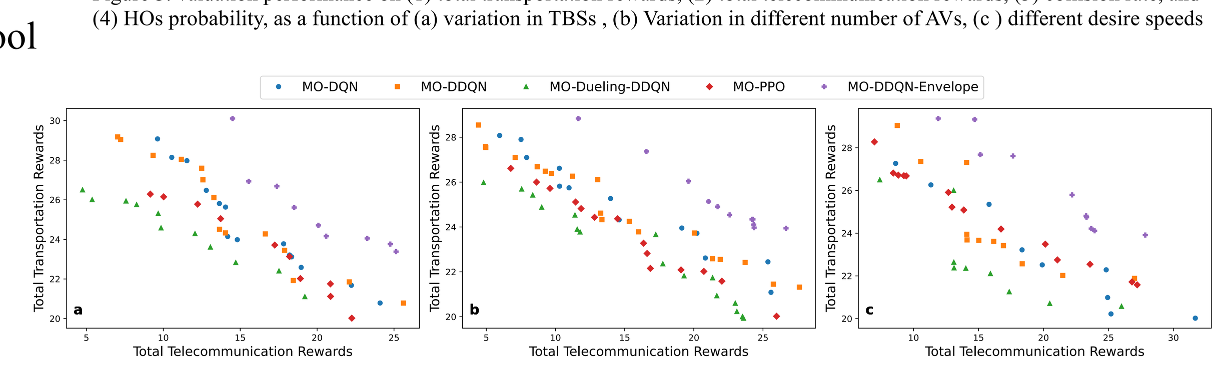


Figure 6: Pareto Frontier Comparison in MOO for total Transportation reward and total telecommunication reward among MO-DQN, MO-DDQN, MO-dueling-DDQN, MO-PPO, and MO-DDQN-Envelop, across instances:(a) I-(20,30,20,20), (b) I-(20,30,10,20), (c) I-(20,30,20,50)

## References


- Yan, Z., & Tabassum, H. (2024). Generalized Multi-Objective Reinforcement Learning with Envelope Updates in URLLC-enabled Vehicular Networks. *arXiv preprint arXiv:2405.11331*.
- Yan, Z., & Tabassum, H. (2022, December). Reinforcement learning for joint V2I network selection and autonomous driving policies. In *GLOBECOM 2022-2022 IEEE Global Communications Conference* (pp. 1241-1246). IEEE.
- Yang, R., Sun, X., & Narasimhan, K. (2019). A generalized algorithm for multi-objective reinforcement learning and policy adaptation. *Advances in neural information processing systems, 32*.


**Acknowledgement**

LASSONDE SCHOOL OF ENGINEERING YORK U NSERC CRSNG