V2XFormer: A Multi-Stage Transformer for Multi-Agent Reinforcement Learning in V2X-Enabled Traffic Signal Control

Yifeng Zhang*, Ping Gong*, Weiyi He, Yilin Liu, Mingfeng Fan†, Guillaume Sartoretti

***Abstract*— Connected vehicles (CVs), enabled by vehicle-to-everything (V2X) communication, provide more fine-grained and real-time traffic information, offering new opportunities to enhance network-wide adaptive traffic signal control (ATSC). However, effectively integrating heterogeneous CV and infrastructure data, and leveraging such information for cooperative ATSC across multiple intersections remain critical challenges in CV-enabled traffic environments. To address this, we propose V2XFormer, a multi-stage Transformer framework designed to fuse features from CVs and intersections via vehicle-level temporal encoding, lane-level interaction modeling, and intersection-level coordination modeling, and jointly optimize traffic prediction and MARL-based signal control for cooperative control in CV-enabled environments. Specifically, at the lane level, we design a dual-encoder Transformer that utilizes temporal vehicle information to extract features from cooperative and competitive lanes, and enhance feature aggregation via an adaptive gated fusion mechanism guided by intersection-level context. At the intersection level, we introduce a decoder-only Transformer that adaptively integrates local and neighboring intersection features to enable broader cross-intersection coordination. This hierarchical design allows V2XFormer to capture both fine-grained lane interactions and high-level intersection dependencies, leading to more consistent and stable control policies over time. Experimental results show that our method consistently outperforms various baselines in network-wide traffic optimization, with notable improvements under high-demand and complex shared-lane scenarios, highlighting its effectiveness for large-scale ATSC in CV-enabled environments.**

I. Introduction

With the rapid progress of urbanization and the continuous growth in the number of vehicles, traffic congestion has become an increasingly serious problem in urban areas [1]. Adaptive Traffic Signal Control (ATSC) has been widely recognized as an effective way to reduce delays, shorten travel time, and improve overall driving experience [2]. In recent years, data-driven methods, especially Reinforcement Learning (RL) approaches, have attracted increasing attention in ATSC [3], [4], [5]. Through interaction with the environment, RL agents can learn signal control strategies that effectively balance long-term traffic optimization with short-term adaptation to dynamic traffic conditions. Overall, data-driven ATSC is becoming an important component of Intelligent Transportation Systems (ITS), contributing to smarter and more responsive traffic control in urban areas.

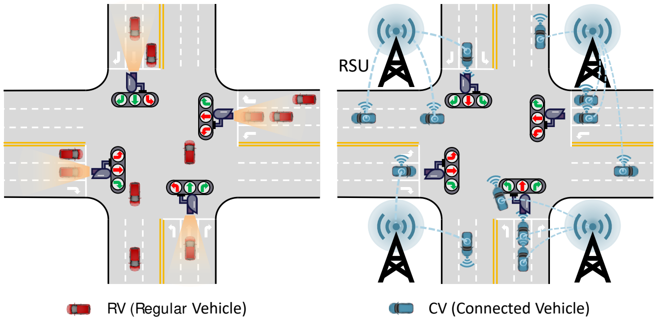


Fig. 1: An illustration of different ATSC frameworks in the traditional regular vehicles (RVs) environment (left) and the connected vehicles (CVs) environment (right).

Most existing studies focus on traffic environments that mainly consist of regular vehicles (RVs), where ATSC systems mainly rely on roadside sensors, such as cameras and inductive loop detectors, to collect real-time traffic data (e.g., vehicle counts). Decisions are usually made based on local traffic conditions at each intersection, as shown on the left side of Fig. 1. However, these systems still face challenges in complex real-world scenarios due to the limited view, fixed position, and inability of roadside sensors to capture intention-level information. For example, heavy traffic in shared lanes often limits accurate observations due to occlusion and the difficulty in inferring vehicle intentions, while roadside sensors may also fail under adverse weather or lighting conditions [6]. As a result, traditional RL methods that rely solely on low-dimensional local observations often struggle to capture overall traffic conditions, leading to suboptimal strategies. In future CV environments, vehicles can share real-time information about their status and driving intentions through Roadside Units (RSUs) with V2X communication [7], [8], allowing signal control to rely on more fine-grained and high-dimensional observations at intersections, as shown on the right side of Fig. 1. However, integrating data from both vehicles and infrastructure, and using fused features to enable high-quality cooperative decision-making across multiple intersections remain a key challenge for current RL-based ATSC methods.

To address these challenges, we propose V2XFormer, a multi-stage Transformer framework that performs vehicle-level temporal encoding, lane-level interaction modeling, and

* The first two authors contributed equally to this work.

This work is supported by A*STAR, CISCO Systems (USA) Pte. Ltd and National University of Singapore under its Cisco-NUS Accelerated Digital Economy Corporate Laboratory (Award I21001E0002). (*Corresponding author: Mingfeng Fan*).

Yifeng Zhang, Ping Gong, Mingfeng Fan, and Guillaume Sartoretti are with the Department of Mechanical Engineering, National University of Singapore, Singapore (E-mail: {yifeng, e1133090}@u.nus.edu, {ming.fan, guillaume.sartoretti}@nus.edu.sg).

Weiyi He is with the Department of Civil Engineering, National University of Singapore, Singapore (E-mail: e1177407@u.nus.edu).

Yilin Liu is with the State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications (E-mail: liuyilin10@bupt.edu.cn)

intersection-level coordination modeling, to fuse heterogeneous traffic data from both vehicles and infrastructure, and jointly learn prediction and control strategies for cooperative ATSC in V2X-enabled environments. Specifically, at the vehicle level, we first employ a GRU module to encode the historical trajectories of CVs, effectively capturing their dynamic motion patterns. At the lane level, we design a dual-encoder Transformer that leverages the temporal vehicle data to separately extract distinctive features from cooperative (non-conflicting) and competitive (conflicting) lanes at each intersection. To effectively fuse these lane-specific features, we introduce an adaptive gating mechanism that learns to weigh cooperative and competitive information, where the gating behavior is optimized under the joint supervision of both prediction and RL objectives, enabling more effective task-driven feature fusion. Next, we decode the fused features using intersection-level traffic observations, explicitly modeling interactions between CVs and intersections to obtain a unified representation of the overall traffic state. At the intersection level, we further incorporate a decoder-only Transformer that adaptively aggregates features from both local and neighboring intersections to facilitate inter-intersection cooperation. In contrast to traditional methods that address prediction and control separately, V2XFormer employs a unified model to jointly optimize future traffic prediction and MARL-based signal control, promoting more consistent task-aligned representation learning and enabling forward-looking decision-making, which together improve the robustness and effectiveness of the learned policy.

We evaluate V2XFormer on the open-source *Grid 5×5* dataset [9], which includes 25 four-arm intersections with different traffic demands (e.g., low, medium, and high) and shared-lane complexity (i.e., *Grid 5×5 V2*). Experimental results show that V2XFormer outperforms existing ATSC methods in reducing traffic congestion, emissions, and fuel consumption, especially in more challenging shared-lane scenarios. We believe this stems from V2XFormer's ability to effectively integrate CV information, especially their driving intentions, with concise intersection-level traffic states, enabling more informed decisions in such complex settings. Visualized CV trajectories with time also show that our method can effectively reduce delays and stop-and-go behaviors. These results highlight the advantage of our framework in fusing fine-grained CV and intersection information, and demonstrate its potential as a practical solution for collaborative ATSC in CV-enabled environments.

II. Related Work

In RV environments, traditional TSC methods typically relied on roadside sensors to detect traffic conditions. Fixed-time methods, such as the Webster method [10], predefined signal plans based on historical data, and cannot adapt well to real-time traffic variations. To improve adaptability, adaptive systems like SCOOT [11] and SCATS [12] adjusted signal timings using real-time sensor inputs. The max-pressure method [13] further optimized traffic flow by adjusting signal phases to balance pressures between upstream and downstream intersections. In more complex and highly dynamic urban environments, MARL has emerged as a key approach for ATSC [2]. For homogeneous networks, PressLight [14] introduced the concept of "pressure" into the state and reward definitions, guiding agents to select phases based on higher pressure. Methods such as CoLight [3], STMARL [15], and GPLight [5] employed graph neural networks (GNNs) to capture spatial and temporal relationships among intersections, thereby implicitly enhancing cooperation among agents. SocialLight [4] proposed a distributed RL framework using local counterfactual reasoning to estimate each agent's contribution within the neighborhood, thus improving policy stability and scalability. CoordLight [16] introduced a novel state representation called QDSE to improve local traffic dynamics modeling and employed an attention mechanism to enhance coordination among key neighbors, thereby enabling network-level optimization. For general ATSC methods in heterogeneous networks, FRAP [17] proposed a phase competition mechanism to handle diverse intersections. GESA [18] extended this by introducing a scenario-agnostic RL framework, which included a rule-based module to unify intersection representations without manual labeling, enabling simultaneous training across multiple scenarios. Unicorn [19] proposed a unified state-action representation method, leveraging cross-attention and contrastive learning to capture both common and unique features across intersections, along with a collaborative optimization strategy for scalable and generalizable network-wide ATSC.

In CV environments, ATSC methods benefit from advanced V2X communication technologies, providing richer and more accurate real-time traffic information compared to roadside sensors. Non-learning-based methods often formulate TSC as optimization problems using data from CVs. These methods [6], [20] usually apply mixed-integer linear programming (MILP) or dynamic programming (DP) to optimize signal phases by estimating detailed vehicle trajectories or aggregate flow measures like queue lengths and densities. Recent V2X-based methods have explored different ways to improve ATSC performance using data from CVs. Early study [21] showed that using detailed V2I data like vehicle positions and speeds can greatly improve traffic flow and reduce delays. CVLight [7] proposed a decentralized RL method that combines CV and non-CV data during training, enabling stable control even with low CV penetration. Later, UniTSA [8] introduced a universal RL framework for V2X environments, using a unified state design and data augmentation to adapt to different intersection types. Pang et al. [22] proposed a two-stage RL method that predicts traffic and refines actions using real-time V2X data for edge-computing scenarios. Despite these efforts, there is still a lack of effective and general solutions for fusing heterogeneous CV and intersection data and using them to coordinate multiple intersections in CV-enabled environments.

III. Background

A. Traffic Terminology

We first introduce the traffic terminology used in this work.

- ***Roads and lanes***: Incoming roads are where vehicles enter an intersection, while outgoing roads are where vehicles exit. Each road contains one or multiple lanes.
- ***Traffic movement and traffic phase***: A traffic movement refers to the vehicle flow from a incoming lane to a specific outgoing lane. A traffic phase is a set of allowed traffic movements that can proceed simultaneously without conflict, typically controlled by traffic signals.
- ***Traffic agent and traffic network***: A traffic agent is a signal controller that makes decisions such as selecting signal phases and timings for a specific intersection. A traffic network is a connected system of intersections and roads where multiple agents operate and interact.
- ***Traffic neighborhood***: A traffic neighborhood refers to an agent and all intersections/agents directly connected to it, where the set of neighbors is denoted by $\mathcal{N}$.

B. TSC as MARL

To capture the partial observability and decentralized nature of the network-wide TSC problem, we formulate it as a Decentralized Partially Observable Markov Decision Process (Dec-POMDP) [23], defined as $G = (N, S, A, R, T, O, \rho, \gamma)$. In this formulation, each intersection is modeled as a learning agent that makes decisions based on its own local observations and interacts with neighbors through the environment. Here, N is the number of agents, S is the global state space, and $A = [A_1, A_2, \ldots, A_N]$ is the joint action space. The reward function is given by $R : S \times A \times S \to R$, and the state transition function $T(s^{'}|s, a) : S \times A \times S \to [0, 1]$ defines the probability of transitioning to state s' after taking action a in state s. Since agents cannot directly observe the global state, each agent i receives a local observation $z_i \in Z$, determined by the observation function $O_i(s) : S \to Z$. In this work, we consider a fully CV-enabled environment, where each agent's observation includes both the macroscopic traffic state of its own intersection and microscopic real-time state information from individual CVs. Additionally, ρ denotes the initial state distribution, and γ is the discount factor. Given a joint policy π, the objective is to find the optimal policy π^* that maximizes the cumulative discounted reward for all agents: $G(\pi) = E_{\tau \sim \pi}\left[\sum_{i=1}^{N} \sum_{t=0}^{\tau_{len}} \gamma^t r_i^t\right]$, where τ denotes the global state-action trajectory and τ_{len} is the time horizon.

C. RL Agent Design

1) Observation: In a CV environment, each RL agent (i.e., intersection) constructs its observation from two components: ***vehicle-level observations*** obtained via V2X communication from CVs and ***intersection-level observations*** that reflect the overall local traffic conditions. For each incoming lane l_{in}, where $\mathcal{L}_{in}$ is the set of all incoming lanes at the intersection, the agent collects information from CVs currently on that lane via RSUs. The set of CVs on lane l_{in} is denoted as $\mathcal{V}_{l_{in}}$, and their observations are represented as $o_{cv}^{l_{in}} = [o_v,\, v \in \mathcal{V}_{l_{in}}]$, where each vehicle v is characterized by $o_v = (p_v, s_v, a_v, d_v)$, including its lane position p_v, speed s_v, acceleration a_v, and intended route d_v. The intended route d_v is encoded as a one-hot vector indicating the vehicle's next target intersection. By aggregating all CV observations across incoming lanes, the agent forms a structured vehicle-level observation: $o_{cv} = [o_{cv}^{l_{in}},\, l_{in} \in \mathcal{L}_{in}]$. In parallel, the agent constructs an ***intersection-level observation*** $o_{int} = (s_{int}, n_{int}^{move}, n_{int}^{wait})$, where s_{int} is a one hot vector indicating the current activated signal phase, n_{int}^{move} is the number of moving vehicles per incoming lane, and n_{int}^{wait} is the number of stopped vehicles per incoming lane. This hybrid observation design $o = (o_{cv}, o_{int})$ enables each agent to perceive both fine-grained vehicle-level dynamics and summarized intersection-level traffic states, aiming to facilitate more informed and context-aware decision-making. To further broaden each agent's field of view, it also receives observations from neighboring intersections, allowing its policy to account for both local and adjacent traffic conditions.

2) Action: We define the action as selecting a signal phase that remains active for a fixed period (e.g., 5 seconds) until the next decision step. For a typical four-way intersection, there are eight valid traffic phases, which include *north-south straight*, *east-west straight*, *north-south left turn*, *east-west left turn*, *north straight and left turn*, *south straight and left turn*, *east straight and left turn*, and *west straight and left turn*, while right turns are always permitted and not controlled by the phases. A yellow light (e.g., 2 seconds) is added before switching phases to ensure safety.

3) Reward: Aligning with prior work [9], [4], [16], we adopt queue length (i.e., stopped vehicle count) as the reward signal, where the local reward for each agent is defined as $r = -\frac{1}{|\mathcal{L}_{in}|} \sum_{l_{in} \in \mathcal{L}_{in}} n_{l_{in}}^{\text{wait}}$, where $n_{l_{in}}^{\text{wait}}$ is the number of stopped vehicles on lane l_{in}. To promote better coordination and cooperation among neighboring intersections and improve overall network performance, the final reward for each agent i is computed by averaging the rewards of itself and its neighbors $\mathcal{N}_i$: $\hat{r}_i = \frac{1}{|\mathcal{N}_i \cup \{i\}|} \sum_{j \in \mathcal{N}_i \cup \{i\}} r_j$.

IV. V2XFormer

In this section, we first present the network architecture of V2XFormer. Following this, we introduce a joint optimization framework that combines future traffic prediction and MARL-based control into a unified learning model, enabling more effective ATSC with improved long-term performance.

A. Network Architecture

In V2XFormer, we process and fuse heterogeneous data from CVs and intersections through a multi-stage network structure composed of three levels—vehicle, lane, and intersection, as illustrated in Fig. 2. For each intersection, we first construct the vehicle-level observation represented as $o_{cv} \in \mathbb{R}^{|\mathcal{L}_{in}| \times d_l} = [o_{cv,l_{in}}, l_{in} \in \mathcal{L}_{in}]$, as described in Sec. III-C.1. Since the number of CVs varies across lanes over time, we apply a padding mechanism to ensure a fixed observation dimension $d_l = 210$ for each lane. The observation o_{cv} is then projected into a higher-dimensional feature using a two-layer MLP (multilayer perceptron): $h_{cv} \in \mathbb{R}^{|\mathcal{L}_{in}| \times d_h} = \mathrm{MLP}\,(o_{cv})$, where d_h is the hidden feature dimension. To incorporate temporal dependencies, we apply a gated recurrent unit (GRU) module [24] that captures the

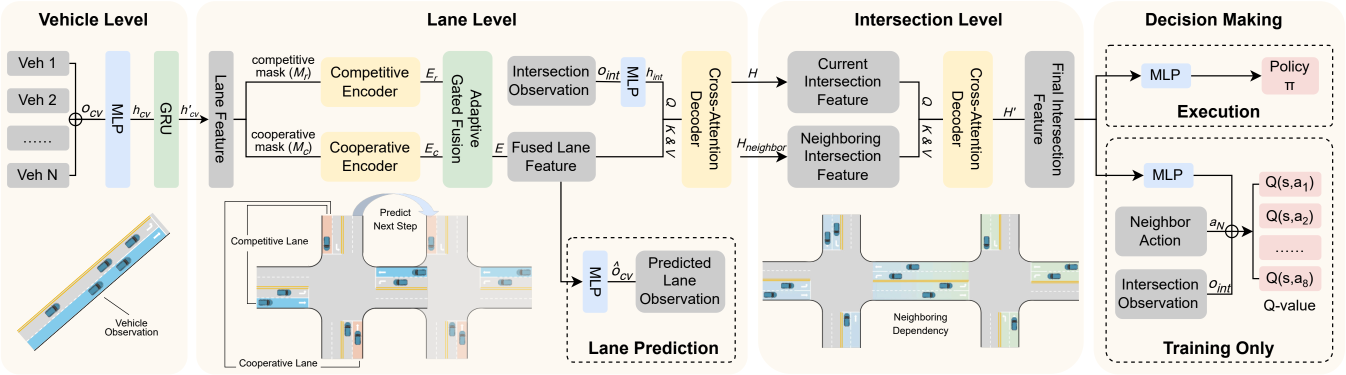


Fig. 2: Overview of the proposed V2XFormer framework, designed with a multi-stage Transformer to fuse heterogeneous features and jointly optimize traffic prediction and MARL-based control for cooperative ATSC in CV-enabled environments.

historical dynamics of CVs at each lane, leading to the feature h'_{cv}: $h'_{cv} = \mathrm{GRU}\left(h_{cv}\right)$. Modeling these historical dynamics is crucial as it helps improve traffic estimation and enables more adaptive and efficient signal control.

Next, we feed the temporal features $h_{cv}^{'}$ into a dual-encoder Transformer, consisting of two parallel self-attention encoders [25] with different masking mechanisms to model cooperative and competitive relationships between different incoming lanes. This design enables the model to distinguish between lanes that can move together and those that compete for the same signal phase, leading to more accurate modeling of their interactions. The *cooperative encoder* applies a cooperative mask M_c, where lanes that conflict have attention values of $-\infty$, preventing information aggregation, while non-conflicting lanes retain their original attention values. Conversely, the *competitive encoder* applies a competitive mask M_r, where non-conflicting lanes are masked with $-\infty$, ensuring that only conflicting lanes are attended to. The self-attention mechanism for the encoders is computed as follows:

$$\mathrm{Attention}(Q, K, V, M) = \mathrm{softmax}\left(\frac{QK^T}{\sqrt{d}} + M\right) V, \quad (1)$$

where query, key, and value matrices are derived from h'_{cv} through linear transformations: $Q = W_Q\, h'_{cv}$, $K = W_K\, h'_{cv}$, and $V = W_V\, h'_{cv}$, with W_Q, W_K, and W_V being the respective learnable weight matrices for these transformations. The outputs of the encoders represent cooperative features $E_c \in \mathbb{R}^{|\mathcal{L}_{in}| \times d_h}$ and competitive features $E_r \in \mathbb{R}^{|\mathcal{L}_{in}| \times d_h}$:

$$E_c = \mathrm{Encoder}_{l_1}\left(h'_{cv}, M_c\right), \quad E_r = \mathrm{Encoder}_{l_2}\left(h'_{cv}, M_r\right). \quad (2)$$

To adaptively integrate cooperative and competitive features, we introduce an adaptive gated fusion mechanism, where a gating value is first computed using a linear transformation followed by a sigmoid activation: $g = \sigma\left(W_g\, E_c + b_g\right)$. The final fused feature $E \in \mathbb{R}^{|L| \times d_h}$ is then obtained by weighting the cooperative and competitive features: $E = g \cdot E_c + (1 - g) \cdot E_r$. To combine the fused lane-level features with high-level intersection context, we use a cross-attention decoder. We first apply a two-layer MLP to the intersection-level observation o_{int} to get a feature h_{int}: $h_{int} \in \mathbb{R}^{1 \times d_h} = \mathrm{MLP}\left(o_{int}\right)$. Then, we compute the query, key, and value for the cross-attention mechanism similar to Eq. 1, where the query is projected from h_{int}, and the key and value are projected from the fused lane-level features E. This allows the model to align lane-level dynamics with high-level intersection features, leading to a refined and context-aware representation: $H \in \mathbb{R}^{1 \times d_h} = \mathrm{Decoder}_l(h_{int}, E)$.

To build on the current intersection representation and further capture inter-intersection dependencies, we add a decoder-only Transformer (shown as *Cross-Attention Decoder* in Fig. 2) that aggregates critical information from adjacent intersections using the same cross-attention mechanism, resulting in an updated intersection feature denoted as H'. Here, the query is derived from the current intersection's fused feature, while the keys and values are computed from the fused features of its neighboring intersections:

$$H' = \mathrm{Decoder}_{int}\left(H, H_{\text{neighbor}}\right), \quad (3)$$

where $H_{neighbor}$ denotes the features of neighboring intersections obtained through the same encoding process. This enables the model to account for broader neighborhood coordination and cooperation when making decisions. By integrating vehicle-level historical modeling, cooperative-competitive lane-level encoding, and inter-intersection dependency modeling, our framework effectively captures traffic interactions at multiple spatial and temporal scales, enabling more informed and coordinated control decisions. The resulting intersection representation is then used for both prediction and decision-making in the optimization process.

B. Optimization Algorithm

To unify traffic forecasting and control in CV-enabled ATSC, we propose a joint learning paradigm that integrates supervised prediction and MARL-based decision-making. Instead of directly using predicted results for decision-making, we introduce a prediction loss as an auxiliary objective to guide the learning of control policies. The key idea is to use a prediction loss to guide the RL process through shared gradients during training, enabling the policy to better anticipate and react to future traffic changes. For the traffic prediction process, we utilize the fused feature E, extracted from the dual-encoder Transformer of V2XFormer, to predict the vehicle-level observation for the next decision step. This is achieved by passing E through a two-layer MLP, which generates the predicted lane observation $\hat{o}_{cv}^t$. The prediction loss function is defined as the mean squared error (MSE) between the predicted observation $\hat{o}_{cv}^t$ and the ground truth observation at next step o_{cv}^{t+1}: $\mathcal{L}_{\text{pred}} = \text{MSE}\left(\hat{o}_{cv}^t, o_{cv}^{t+1}\right)$.

During RL-based optimization, the final aggregated feature H' is passed through two separate MLPs to generate the policy function and the state-action value function. At each step t, we denote the input to the policy as: $\tilde{o}^t = (o^t, m, o_{\mathcal{N}}^t, m_{\mathcal{N}})$ where o^t and m represent the observation and attention masks of the target intersection, and $o_{\mathcal{N}}^t$, $m_{\mathcal{N}}$ are the corresponding observations and masks from neighboring intersections. Then, the policy and value functions, parameterized by ϕ and θ, are defined as: $\pi_\theta(\tilde{o})$, $Q_\phi(\tilde{o}, a, a_{\mathcal{N}})$, where a^t is the action taken by the target intersection and $a_{\mathcal{N}}^t$ are the actions of its neighbors. To facilitate scalable learning of decentralized cooperative policies, we adopt the counterfactual-based advantage formulation introduced in SocialLight [4]. This approach allows each agent to marginalize over the actions of its neighbors and reason about their contributions, leading to more effective coordination in multi-agent settings. The one-step advantage function is defined as:

$$\begin{aligned} A^t = r^t &+ \sum_{a^{t+1}} \pi_\theta(a^{t+1} \mid \tilde{o}^{t+1})\, Q_\phi(\tilde{o}^{t+1}, a^{t+1}, a_{\mathcal{N}}^{t+1}) \\ &- \sum_{a^t} \pi_\theta(a^t \mid \tilde{o}^t)\, Q_\phi(\tilde{o}^t, a^t, a_{\mathcal{N}}^t). \end{aligned} \quad (4)$$

To further reduce variance and improve training stability, we apply Generalized Advantage Estimation (GAE) [26]. The final advantage is computed as a discounted sum of future one-step advantages: $\hat{A}^t = \sum_{l=0}^{\tau_{len}-t-1} (\gamma\lambda)^l A^{t+l}$, where γ is the discount factor and λ is the GAE smoothing parameter.

To ensure stable policy updates, we apply the Proximal Policy Optimization (PPO) algorithm [27], which uses a clipped surrogate objective for calculating the policy loss:

$$\mathcal{L}_\pi(\theta) = -\mathbb{E}_\tau \left[\min\left(\rho \hat{A},\, \text{clip}(\rho, 1-\epsilon, 1+\epsilon)\, \hat{A} \right) \right], \quad (5)$$

where ρ is the ratio between the probabilities of the new and old policies, given by $\rho(\theta) = \frac{\pi_\theta(a|\tilde{o})}{\pi_{\theta_{\text{old}}}(a|\tilde{o})}$. The clipping parameter ϵ prevents the policy from changing too much in one update. To encourage exploration, we include an entropy bonus that promotes a higher degree of uncertainty in the policy distribution. The entropy loss is defined as:

$$\mathcal{L}_{\text{entropy}}(\theta) = -\mathbb{E}_\tau \left[\sum_a \pi_\theta(a \mid \tilde{o}) \log \pi_\theta(a \mid \tilde{o}) \right]. \quad (6)$$

The value function is then trained by minimizing the MSE between the Q-value of the current action and a bootstrapped target constructed from the immediate reward and the next-step expected Q-value: $\hat{Q}^t = \hat{r}^t + \sum_{a^{t+1}} \pi_\theta(a^{t+1} \mid \tilde{o}^{t+1})\, Q_\phi(\tilde{o}^{t+1}, a^{t+1}, a_{\mathcal{N}}^{t+1})$. The value loss is defined as:

$$\mathcal{L}_v(\phi) = \mathbb{E}_\tau \left[\left(Q_\phi(\tilde{o}^t, a^t, a_{\mathcal{N}}^t) - \hat{Q}^t \right)^2 \right]. \quad (7)$$

The total RL loss is computed as the weighted sum of three components: $\mathcal{L}_{\text{RL}} = \mathcal{L}_\pi + c_1 \mathcal{L}_v - c_2 \mathcal{L}_{\text{entropy}}$, where c_1 and c_2 are weights for the value and entropy losses, respectively.

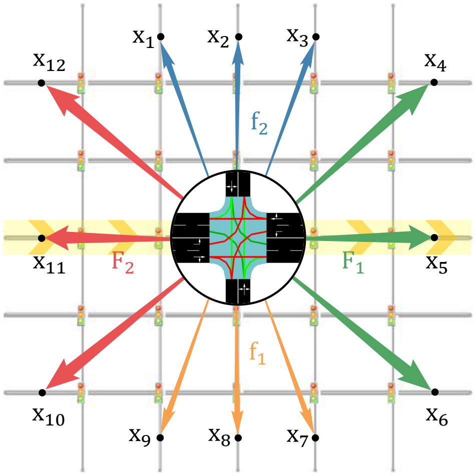


Fig. 3: Illustration of the Grid 5×5 traffic network with 25 intersections and its synthetic major and minor traffic (i.e., CVs) flows, which are adopted from the MA2C paper [9].

Finally, the overall loss function for V2XFormer combines the prediction loss and RL loss: $\mathcal{L} = \mathcal{L}_{\text{RL}} + c_3 \mathcal{L}_{\text{pred}}$, where c_3 is the weight factor. By jointly optimizing the prediction and control objectives, the model can better capture the future traffic trends and learn more forward-looking control strategies, leading to more stable and adaptive system behavior.

V. Experiments and Results

A. *Experimental Settings*

We conduct training and evaluation in the open-source traffic platform SUMO [28]. As the main benchmark, we adopt the *Grid 5×5* traffic dataset from the MA2C paper [9], which consists of 25 homogeneous intersections, as shown in Fig. 3. In this grid network, each intersection has one incoming and one outgoing lane in the north-south direction and two incoming and two outgoing lanes in the east-west direction. To simulate dynamic urban traffic conditions, four synthetic traffic flows that change over time are designed, as represented by the solid and dashed arrows in Fig 3. At the start, three main flows F_1 follow the O-D pairs $x_{10} - x_4$, $x_{11} - x_5$, $x_{12} - x_6$, while three minor flows f_1 move along $x_1 - x_7$, $x_2 - x_8$, and $x_3 - x_9$. After 15 minutes, traffic demands in F_1 and f_1 begin to decrease, and new flows F_2 and f_2 with reversed O-D pairs emerge. The high-demand scenario is set with peak traffic demands of 1100 veh/h for major flows and 925 veh/h for minor flows. For comparison, the low-demand scenario has traffic demands of 220 veh/h and 185 veh/h, while the medium-demand scenario is configured at 660 veh/h and 555 veh/h, respectively. To further test the effectiveness of our method in more challenging shared-lane scenarios, we create another dataset called *Grid 5×5 V2*. The main difference is that each intersection has only one incoming lane in each direction, so vehicles turning left, going straight, or turning right all use the same lane. All other settings are the same as in the original *Grid 5×5* dataset. All experiments were conducted on an Ubuntu server with 32GB RAM, an Intel Core i9-13900KF processor and an NVIDIA RTX 4090 GPU.

TABLE I: Evaluation Performance of different methods under low, medium, and high traffic demand scenarios on the Grid 5×5 network, with the best result in bold and the second-best underlined.

Metrics	Queue Length↓ (vehs)	Speed↑ (m/s)	Trip Completion Rate↑ (vehs/s)	Trip Delay↓ (s)	Fuel Consumption↓ (g/s)	CO2 Emissions↓ (g/s)	Rate of Stop-and-Go↓ (stops/s)
Grid 5x5 (Low Traffic Demand)							
Max-Pressure [13]	0.03(0.03)	7.93(3.28)	0.22(0.22)	22.16(9.16)	40.87(29.08)	128.15(91.18)	0.94(2.09)
PressLight [14]	0.04(0.03)	7.93(3.31)	**0.22(0.21)**	26.22(20.17)	40.09(28.92)	125.70(90.66)	0.73(1.64)
SocialLight [4]	0.02(0.01)	8.98(3.81)	**0.22(0.21)**	9.90(10.04)	35.61(25.66)	111.64(80.44)	0.51(1.24)
SocialLight-CV	**0.00(0.01)**	**10.19(4.48)**	**0.22(0.21)**	2.18(3.24)	**29.12(21.63)**	**91.30(67.83)**	**0.15(0.43)**
V2XFormer(ours)	**0.00(0.01)**	9.92(4.24)	**0.22**(0.22)	**2.04(2.83)**	31.47(22.63)	98.66(70.95)	0.17(0.47)
Grid 5x5 (Medium Traffic Demand)							
Max-Pressure [13]	0.74(0.64)	4.74(2.15)	0.62(0.42)	170.48(226.56)	243.68(163.87)	763.97(513.76)	7.52(6.40)
PressLight [14]	0.36(0.32)	5.77(2.45)	**0.63(0.49)**	85.33(86.78)	175.99(133.46)	551.49(418.42)	4.82(5.03)
SocialLight [4]	0.10(0.10)	7.89(3.24)	**0.63**(0.51)	23.12(21.85)	122.03(92.54)	382.60(290.13)	2.59(3.71)
SocialLight-CV	0.05(0.06)	8.86(3.65)	**0.63**(0.51)	11.83(13.01)	107.42(80.71)	336.79(253.04)	1.57(2.67)
V2XFormer(ours)	**0.04(0.05)**	**8.97(3.68)**	**0.63**(0.51)	**9.20(16.52)**	**107.26(79.20)**	**336.30(248.30)**	**1.21(2.13)**
Grid 5x5 (High Traffic Demand)							
Max-Pressure [13]	5.80(3.89)	1.98(2.62)	0.31(0.27)	340.22(489.08)	881.76(445.23)	2764.36(1395.80)	12.48(6.07)
PressLight [14]	2.98(1.77)	3.02(2.02)	0.90(0.40)	440.12(455.96)	650.98(305.59)	2040.90(958.06)	15.33(7.55)
SocialLight [4]	1.99(1.28)	3.55(2.23)	0.92(0.42)	296.34(467.02)	529.47(244.33)	1659.96(766.00)	14.10(7.82)
SocialLight-CV	1.07(0.91)	4.84(2.20)	**1.03(0.56)**	154.50(208.34)	390.92(241.11)	1225.60(755.90)	10.47(8.16)
V2XFormer(ours)	**0.77(0.74)**	**5.32(2.58)**	**1.03**(0.61)	**111.50(165.93)**	**335.42(224.82)**	**1051.59(704.85)**	**8.19(7.06)**

TABLE II: Evaluation Performance of different methods under low, medium, and high traffic demand scenarios on the Grid 5×5 V2 network, with the best result in bold and the second-best underlined.

Metrics	Queue Length↓ (vehs)	Speed↑ (m/s)	Trip Completion Rate↑ (veh/s)	Trip Delay↓ (s)	Fuel Consumption↓ (g/s)	CO2 Emissions↓ (g/s)	Rate of Stop-and-Go↓ (stops/s)
Grid 5x5 V2 (Low Traffic Demand)							
PressLight [14]	0.07(0.06)	7.79(3.29)	**0.22**(0.22)	30.48(21.54)	41.65(30.43)	130.58(95.42)	0.77(1.75)
SocialLight [4]	0.03(0.02)	8.99(3.80)	**0.22(0.21)**	10.73(10.19)	35.68(25.83)	111.88(80.97)	0.48(1.18)
SocialLight-CV	**0.01(0.01)**	**10.08(4.44)**	**0.22(0.21)**	2.61(3.49)	**30.03(22.47)**	**94.16(70.44)**	**0.18(0.49)**
V2XFormer(ours)	**0.01(0.01)**	9.75(4.19)	**0.22**(0.22)	**2.46(4.54)**	32.85(23.71)	102.98(74.33)	0.22(0.57)
Grid 5x5 V2 (Medium Traffic Demand)							
PressLight [14]	1.41(1.11)	4.53(2.44)	0.61(0.41)	225.35(270.53)	274.23(178.96)	859.74(561.06)	6.57(5.40)
SocialLight [4]	0.21(0.22)	7.42(3.18)	**0.63**(0.50)	32.92(33.37)	134.95(105.66)	423.11(331.26)	3.27(4.40)
SocialLight-CV	0.09(0.10)	8.66(3.61)	**0.63(0.51)**	13.14(14.74)	111.28(84.20)	348.88(263.99)	1.64(2.61)
V2XFormer(ours)	**0.06(0.07)**	**8.90(3.67)**	**0.63(0.51)**	**8.19(11.13)**	**108.50(80.12)**	**340.16(251.20)**	**1.29(2.21)**
Grid 5x5 V2 (High Traffic Demand)							
PressLight [14]	3.66(1.78)	2.71(1.37)	0.80(0.40)	383.41(476.56)	548.42(205.15)	1719.36(643.17)	11.35(4.07)
SocialLight [4]	3.84(2.40)	2.92(2.52)	0.68(0.31)	432.82(565.26)	554.18(228.86)	1737.42(717.50)	12.20(5.37)
SocialLight-CV	2.62(1.69)	4.03(2.53)	0.93(0.40)	258.98(399.98)	474.25(209.05)	1486.85(655.39)	10.12(5.57)
V2XFormer(ours)	**2.14(1.46)**	**4.57(2.53)**	**0.97(0.43)**	**208.24(317.53)**	**428.86(196.41)**	**1344.54(615.77)**	**9.18(5.49)**

B. Evaluation Baselines and Metrics

We compare our approach with three types of baselines: rule-based methods, RL-based ATSC methods, and CV-enhanced RL-based methods, presented as follows:

1) **Rule-based TSC methods**: Max-Pressure [13].
2) **RL-based TSC methods**: PressLight [14] and SocialLight [4]. These methods adopt a parameter-sharing scheme, where a single shared policy is trained using trajectories collected from multiple agents to improve scalability and adaptability across the traffic network.
3) **CV-enabled RL-based methods**: SocialLight-CV, a variant that extends existing RL-based methods by incorporating CV state information as additional input, processed by simple MLPs for decision-making.

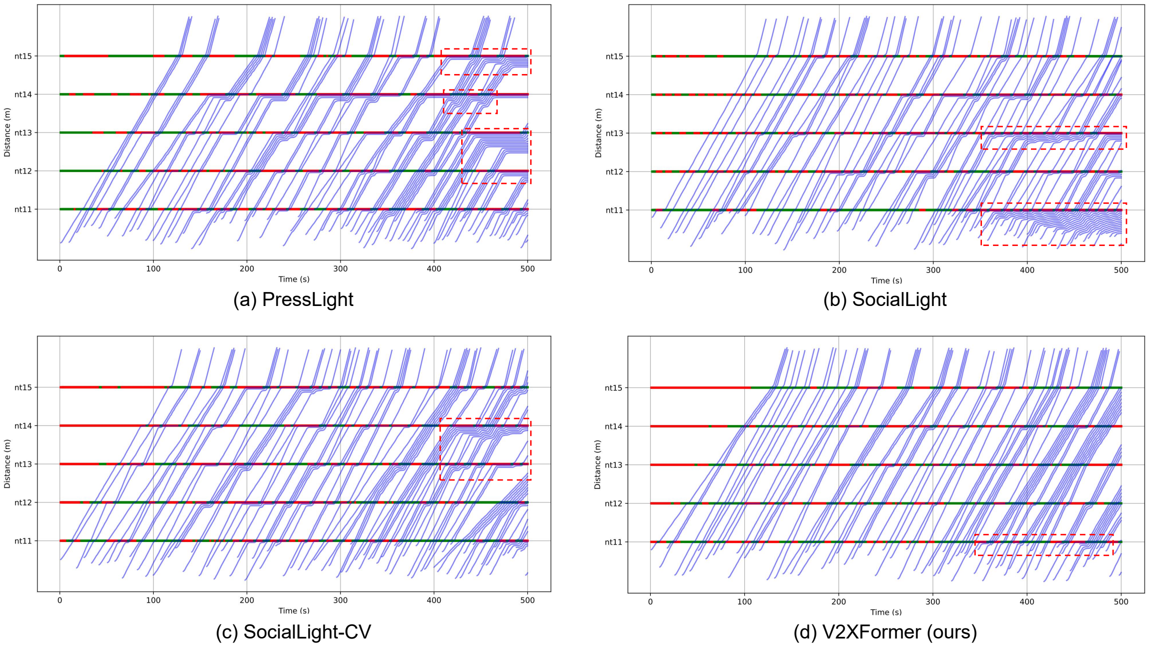
To comprehensively evaluate traffic performance of traffic performance, environmental impact, and safety across different methods, we adopt the following traffic metrics:

1) **Average queue length (veh/s)**: Average number of vehicles waiting at an intersection for each step.
2) **Average vehicle speed (m/s)**: Average speed of all vehicles traveling within the network at each step.
3) **Trip completion rate (veh/s)**: Rate at which vehicles complete their trips, defined as the number of vehicles reaching their destination per second.
4) **Average trip delay (s)**: Average time a vehicle is delayed during its trip compared to the ideal trip time. It is calculated as the difference between the actual trip time and the ideal trip time for completed trips.
5) **Fuel consumption (mg/s)**: Amount of fuel used by vehicles in the network per second. In our experiments, fuel consumption, along with CO_2 emissions (described below), is calculated using the HBEFA3/PC G EU4 model [29]. This model simulates a gasoline-powered Euro 4 standard passenger car and is the default model in the SUMO simulator[1].
6) CO_2 **emissions (mg/s)**: Amount of carbon dioxide produced by vehicles within the network per second. CO_2 emissions are influenced by multiple factors such as speed, acceleration, and fuel consumption.
7) **Rate of stop-and-go (stop/s)**: Average number of stop-and-go events per second across all vehicles in the traffic network. A stop-and-go event is recorded when a vehicle's speed falls below a fixed threshold (set to 0.1 m/s) and then increases again.

C. Results and Analysis

To evaluate the effectiveness of our approach, we compare V2XFormer with several advanced RL-based ATSC methods,

[1] https://sumo.dlr.de/docs/Models/Emissions.html



(a) PressLight

(b) SocialLight

(c) SocialLight-CV

(d) V2XFormer (ours)

Fig. 4: Spatiotemporal trajectories of CVs traveling from west to east during the first 500 seconds in the *Grid 5×5* network. Regions where vehicles stopped and queued are highlighted with red dashed boxes.

including SocialLight-CV, SocialLight, and PressLight. We run experiments on two network setups: *Grid 5×5* and *Grid 5×5 V2* (only one incoming lane per direction), under three traffic demand levels: low, medium, and high. Table I shows the results on the *Grid 5×5* network. Under low traffic demand, the scenario is simple, and congestion rarely happens. Therefore, the performance differences between methods are small, with average queue lengths mostly between 0.01 and 0.04 vehicles. Even so, V2XFormer achieves the best performance, with the lowest average queue length and shortest trip delay. Under medium traffic demand, performance differences become clearer. V2XFormer consistently outperforms the other methods across all key metrics. In the most challenging high-demand scenario, the performance gap further increases. V2XFormer achieves an average queue length of 0.77 vehicles, much lower than the other methods (all exceeding 1.0 vehicles). Additionally, V2XFormer significantly reduces stop-and-go traffic behavior by integrating fine-grained vehicle-level CV features with compact intersection-level representations. Compared to traditional methods that rely solely on roadside sensors, V2XFormer enables a more responsive and predictive ATSC. In particular, we observed that it enables CVs from crossing directions to naturally form alternating platoons, thereby reducing intersection blocking and promoting smoother traffic flow without prolonged stops. The rule-based Max-Pressure method performs poorly in terms of queue length and trip completion rate. This is mainly because it makes decisions based only on the current pressure without considering long-term performance. As a result, it often fails to make better decisions over time, which limits its control ability in complex traffic scenarios.

In addition, Table II presents the experimental results on the *Grid 5×5 V2* network. Under low traffic demand, although all methods show similar trip completion rates (approximately 0.22 veh/s), V2XFormer demonstrates better performance with the lowest average trip delay (2.46 sec). As traffic demand increases to the medium level, V2XFormer consistently outperforms all baselines across all evaluation metrics, particularly excelling in improving traffic flow and reducing environmental impact. In the more challenging high-demand scenario, the network has fewer lanes and higher traffic density, leading to more entangled vehicle routes and more diverse traffic demands. This setting provides a better test of the model's ability to coordinate different traffic needs and improve overall traffic performance. The results show that SocialLight-CV and V2XFormer, both leveraging CV information, outperform traditional RL-based methods. In particular, V2XFormer adopts a multi-stage Transformer to effectively fuse multi-scale heterogeneous information, enabling the model to better capture dynamic interactions across levels and significantly enhance performance in CV-enabled environments. Furthermore, through vehicle intent sharing, V2XFormer enables agents to make more informed and adaptive phase selection decisions, leading to smoother traffic flow across intersections.

To further show control performance, Fig.4 compares the spatiotemporal trajectories of vehicles under four methods (PressLight, SocialLight, SocialLight-CV, and V2XFormer). The trajectories show vehicle movements from west to east through the middle five intersections (highlighted in Fig.3) during the first 500 seconds of the simulation (out of a total of 3600 seconds). The control strategies are also shown in red and green, where red indicates that east-west traffic is stopped, and green means it is allowed to go. In Fig.4(a), vehicle trajectories under PressLight show clear zigzag patterns, caused by frequent stop-and-go behavior. This reflects that relying only on roadside sensors may learn sub-optimal policies due to limited information. In Fig. 4(b), SocialLight generates smoother trajectories and improves traffic in some areas, but as demand increases, vehicles start to queue at several intersections, leading to local congestion. This suggests that although SocialLight introduces inter-intersection cooperation, the lack of detailed vehicle data limits the accuracy of local control. Fig. 4(c) shows better results with SocialLight-CV, where vehicle trajectories are smoother and more evenly distributed across intersections, indicating that fine-grained CV data enables better balance between local control and regional coordination. Fig.4(d) shows the best result from V2XFormer. Most vehicle trajectories are nearly straight, with little or no waiting behavior. This shows that V2XFormer effectively adjusts signal control strategies based on real-time vehicle demands, ensuring smoother traffic flow and reducing overall network congestion.

VI. Conclusion

In this paper, we propose V2XFormer, a multi-stage Transformer framework for cooperative ATSC in CV-enabled environments. V2XFormer effectively integrates real-time information from vehicles and intersections at the vehicle, lane, and intersection levels, enabling the joint optimization of traffic prediction and MARL-based decision-making through a unified model. Experimental results demonstrate that V2XFormer achieves superior performance, especially in challenging high-demand and shared-lane scenarios, highlighting its effectiveness and potential for large-scale traffic management in V2X-enabled environments.

Our work has several limitations and future directions. First and foremost, we only considered pure CV scenarios. Although V2X technologies are becoming mature, widely deploying CVs and RSUs remains challenging. Thus, extending our approach to mixed traffic environments involving both CVs and RVs, and enabling collaborative perception among CVs to infer the behaviors of RVs under low CV penetration rates, represents an important direction for our future work. Finally, communication delays and packet losses in realistic V2X environments were not considered in this work. Future work could simulate these issues to further enhance the robustness of V2XFormer in real-world applications.

References

[1] INRIX Research, "Inrix 2024 global traffic scorecard," 2024.

[2] A. Haydari and Y. Yılmaz, "Deep reinforcement learning for intelligent transportation systems: A survey," *IEEE Transactions on Intelligent Transportation Systems (T-ITS)*, vol. 23, no. 1, pp. 11–32, 2020.

[3] H. Wei, N. Xu, H. Zhang, G. Zheng, X. Zang, C. Chen, W. Zhang, Y. Zhu, K. Xu, and Z. Li, "Colight: Learning network-level cooperation for traffic signal control," in *Proc. 28th ACM Int. Conf. Inf. Knowl. Manag.*, pp. 1913–1922, 2019.

[4] H. Goel, Y. Zhang, M. Damani, and G. Sartoretti, "Sociallight: Distributed cooperation learning towards network-wide traffic signal control," *arXiv preprint arXiv:2305.16145*, 2023.

[5] Y. Liu, G. Luo, Q. Yuan, J. Li, L. Jin, B. Chen, and R. Pan, "Gplight: Grouped multi-agent reinforcement learning for large-scale traffic signal control.," in *IJCAI*, pp. 199–207, 2023.

[6] Y. Wang, X. Yang, H. Liang, and Y. Liu, "A review of the self-adaptive traffic signal control system based on future traffic environment," *J. Adv. Transp.*, vol. 2018, no. 1, p. 1096123, 2018.

[7] Z. Mo, W. Li, Y. Fu, K. Ruan, and X. Di, "Cvlight: Decentralized learning for adaptive traffic signal control with connected vehicles," *Transportation research part C: emerging technologies*, vol. 141, p. 103728, 2022.

[8] M. Wang, X. Xiong, Y. Kan, C. Xu, and M.-O. Pun, "Unitsa: A universal reinforcement learning framework for v2x traffic signal control," *IEEE Transactions on Vehicular Technology*, 2024.

[9] T. Chu, J. Wang, L. Codecà, and Z. Li, "Multi-agent deep reinforcement learning for large-scale traffic signal control," *IEEE T-ITS*, vol. 21, no. 3, pp. 1086–1095, 2019.

[10] P. Koonce and L. Rodegerdts, "Traffic signal timing manual.," tech. rep., United States. Federal Highway Administration, 2008.

[11] P. Hunt, D. Robertson, R. Bretherton, and M. C. Royle, "The scoot on-line traffic signal optimisation technique," *Traffic Engineering & Control*, vol. 23, no. 4, 1982.

[12] P. Lowrie, "Scats-a traffic responsive method of controlling urban traffic," *Sales information brochure published by Roads & Traffic Authority, Sydney, Australia*, 1990.

[13] P. Varaiya, "Max pressure control of a network of signalized intersections," *Transportation Research Part C: Emerging Technologies*, vol. 36, pp. 177–195, 2013.

[14] H. Wei, C. Chen, G. Zheng, K. Wu, V. Gayah, K. Xu, and Z. Li, "Presslight: Learning max pressure control to coordinate traffic signals in arterial network," in *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*, pp. 1290–1298, 2019.

[15] Y. Wang, T. Xu, X. Niu, C. Tan, E. Chen, and H. Xiong, "Stmarl: A spatio-temporal multi-agent reinforcement learning approach for cooperative traffic light control," *IEEE Transactions on Mobile Computing*, vol. 21, no. 6, pp. 2228–2242, 2020.

[16] Y. Zhang, H. Goel, P. Li, M. Damani, S. Chinchali, and G. Sartoretti, "Coordlight: Learning decentralized coordination for network-wide traffic signal control," *IEEE T-ITS*, pp. 1–16, 2025.

[17] G. Zheng, Y. Xiong, X. Zang, J. Feng, H. Wei, H. Zhang, Y. Li, K. Xu, and Z. Li, "Learning phase competition for traffic signal control," in *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pp. 1963–1972, 2019.

[18] H. Jiang, Z. Li, Z. Li, L. Bai, H. Mao, W. Ketter, and R. Zhao, "Gesa: A general scenario-agnostic reinforcement learning for traffic signal control," *IEEE T-ITS*, 2024.

[19] Y. Zhang, Y. Liu, P. Gong, P. Li, M. Fan, and G. Sartoretti, "Unicorn: A universal and collaborative reinforcement learning approach towards generalizable network-wide traffic signal control," 2025.

[20] M. A. Palash and D. Wijesekera, "Adaptive traffic signal control using cv2x," in *2023 IEEE 98th Vehicular Technology Conference (VTC2023-Fall)*, pp. 1–7, IEEE, 2023.

[21] J. V. Busch, V. Latzko, M. Reisslein, and F. H. Fitzek, "Optimised traffic light management through reinforcement learning: Traffic state agnostic agent vs. holistic agent with current v2i traffic state knowledge," *IEEE Open J. Intell. Transp. Syst.*, vol. 1, pp. 201–216, 2020.

[22] A. Pang, M. Wang, Y. Chen, M.-O. Pun, and M. Lepech, "Scalable reinforcement learning framework for traffic signal control under communication delays," *IEEE Open J. Veh. Technol.*, 2024.

[23] F. A. Oliehoek, C. Amato, *et al.*, *A concise introduction to decentralized POMDPs*, vol. 1. Springer, 2016.

[24] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," *arXiv preprint arXiv:1406.1078*, 2014.

[25] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, vol. 30, 2017.

[26] J. Schulman, P. Moritz, S. Levine, M. Jordan, and P. Abbeel, "High-dimensional continuous control using generalized advantage estimation," *arXiv preprint arXiv:1506.02438*, 2015.

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

[28] P. A. Lopez, M. Behrisch, L. Bieker-Walz, J. Erdmann, Y.-P. Flötteröd, R. Hilbrich, L. Lücken, J. Rummel, P. Wagner, and E. Wießner, "Microscopic traffic simulation using sumo," in *The IEEE International Conference on Intelligent Transportation Systems*, IEEE, 2018.

[29] M. Keller, S. Hausberger, C. Matzer, P. Wüthrich, and B. Notter, "Handbook of emission factors for road transport (hbefa) 3.1," *Quick Reference, INFRAS, Zurich, Switzerland, Tech. Rep. I-20/2009*, 2010.