MacLight: Multi-scene Aggregation Convolutional Learning for Traffic Signal Control

Sunbowen Lee
College of Science
Hubei Province Key Laboratory of System Science in Metallurgical Process,
Wuhan University of Science and Technology
Wuhan, China
bw1863@wust.edu.cn

Hongqin Lyu*
State Key Lab of Processors, Institute of Computing Technology, Chinese Academy of Sciences
University of Chinese Academy of Sciences
Beijing, China
hongqinlyu@163.com

Yicheng Gong†
College of Science
Hubei Province Key Laboratory of System Science in Metallurgical Process,
Wuhan University of Science and Technology
Wuhan, China
gongyicheng@wust.edu.cn

Yingying Sun
College of Science
Wuhan University of Science and Technology
Wuhan, China
sunyingying082@163.com

Chao Deng
School of Automobile and Traffic Engineering, Wuhan University of Science and Technology
Wuhan, China
woec@wust.edu.cn

ABSTRACT

Reinforcement learning methods have proposed promising traffic signal control policy that can be trained on large road networks. Current SOTA methods model road networks as topological graph structures, incorporate graph attention into deep Q-learning, and merge local and global embeddings to improve policy. However, graph-based methods are difficult to parallelize, resulting in huge time overhead. Moreover, none of the current peer studies have deployed dynamic traffic systems for experiments, which is far from the actual situation.

In this context, we propose Multi-Scene Aggregation Convolutional Learning for traffic signal control (MacLight), which offers faster training speeds and more stable performance. Our approach consists of two main components. The first is the global representation, where we utilize variational autoencoders to compactly compress and extract the global representation. The second component employs the proximal policy optimization algorithm as the backbone, allowing value evaluation to consider both local features and global embedding representations. This backbone model significantly reduces time overhead and ensures stability in policy updates. We validated our method across multiple traffic scenarios under both static and dynamic traffic systems. Experimental results demonstrate that, compared to general and domian SOTA methods, our approach achieves superior stability, optimized convergence levels and the highest time efficiency. The code is under https://github.com/Aegis1863/MacLight.

KEYWORDS

Traffic signal control; Multi-scene convolution; Variational autoencoder; Multi-agent reinforcement learning

ACM Reference Format:
Sunbowen Lee, Hongqin Lyu, Yicheng Gong, Yingying Sun, and Chao Deng. 2025. MacLight: Multi-scene Aggregation Convolutional Learning for Traffic Signal Control. In *Proc. of the 24th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2025), Detroit, Michigan, USA, May 19 – 23, 2025*, IFAAMAS, 9 pages.

1 INTRODUCTION

Traffic signal control (TSC) is an important issue in urban management. As the number of vehicles owned by residents increases, the deteriorating traffic conditions have a serious impact on social development. Traffic signal optimization is a low-cost means to alleviate traffic pressure.

The optimization of traffic light timing constitutes a complex nonlinear stochastic problem, as highlighted in [32]. Traditional intelligent control solutions often resort to assumptions or lack of flexibility, such as unlimited lane capacity [24], Christina Diakaki et al. [6] assumes that the traffic flow is uniform, or fail to adapt effectively to dynamic traffic flows [7]. Consequently, the performance may fall short of that achieved by a fixed timing plan meticulously crafted by human experts.

Although mathematical modeling of real traffic systems is very difficult, the emergence of mature traffic simulators can provide interactive environments, which means that model-free methods can be applied. Reinforcement learning (RL) [21–23] provides SOTA solutions in the field of model-free control. Preliminary RL approaches,

*Contribution equal to the first author
†Corresponding author

Proc. of the 24th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2025), Y. Vorobeychik, S. Das, A. Nowé (eds.), May 19 – 23, 2025, Detroit, Michigan, USA. © 2025 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org).

such as Q-learning [27] and its variants, have shown promising results in optimizing TSC. By iteratively learning from the environment, these algorithms can dynamically adjust signal to minimize congestion.

Intelligent control methods for individual traffic lights are very mature [28], but they are inefficient for large road networks. Current research focuses on whether multiple traffic lights can effectively coordinate to achieve effects such as green wave roads. A common approach is to model the road network as a topology graph structure and introduce Graph Attention Networks (GAT) [25], enabling traffic signals to consider both local and neighboring features for comprehensive optimization decisions through feature aggregation. However, current GAT-based approaches [14, 15, 29] are almost used in Deep Q-learning (DQN) [17]. DQN, As an off-policy framework, despite being data-efficient, graph learning and batch learning consume significant time and computational resources. A more critical issue is they are prone to overfitting, leading to policy collapse.

In this case, we consider both local and global characteristics and propose a novel global scene aggregation approach. **Our approach is motivated by two key points: firstly, the ability of decision-making and value evaluation of agents should to be separated.** Thus, we utilize Proximal Policy Optimization (PPO) [19] as the backbone model, which has a value evaluation module and a policy improvement module to process different information respectively. **Secondly**, global scene aggregation does not necessarily require topological graph modeling. Research by Hua Wei et al. [29] indicates that in topological graph modeling scenarios, **each agent considering only one-hop neighbors yields the best results, which limits the agent's understanding of broader states.** Therefore, we aggregate the features of each agent (scene), using convolutional neural networks (CNN) [11] for a latent global representation. Another important reason for not using graph convolutional neural networks (GCNs) is that it is difficult to compute in parallel and apply to the more advanced Actor Critic RL framework. Consequently, our approach is called multi-scene aggregation convolutional learning (MacLight).

Furthermore, we are the first construct **dynamic traffic flow scenario** by using the professional open-source simulator SUMO [13]. It can simulate the change of traffic flow distribution caused by emergency traffic incidents. We incorporate it as a challenging experimental scenario, alongside other general scenarios to test algorithms. Specifically, we can impose emergency speed limits or ban traffic on any road and reroute all vehicles. Vehicles take speed limits or bans into account and choose new routes, similar to what might happen in real life, causing sudden changes in traffic distribution on some roads. This requires agents not only to cope with familiar traffic characteristics but also to have the ability to handle dynamically changing traffic flows. This greatly expands the scope of existing research.

In summary, the contributions of this paper are as follows:

- We construct a dynamic traffic flow simulation scheme to simulate any possible emergency traffic events, greatly expanding the current research space.
- We propose an online-trained variational autoencoder (VAE) based on CNN for global state representation, obtaining a compact and efficient representation from the latent space for downstream learning.
- We integrate global state representation into the value evaluation module of PPO, enabling the algorithm to balance local and macro characteristics, and demonstrating superior performance compared to both general and domain SOTA.

2 RELATED WORK

Customizable simulator. In the field of TSC, the Simulation of Urban MObility (SUMO) simulator is widely used for urban planning and traffic flow simulation. The simulator allows researchers to define any desired traffic flow scenario. Ma and Wu [16] were among the first to utilize SUMO for traffic control simulations, and it has become the main tool for relevant researchers in recent years. Furthermore, SUMO-RL [1, 2] integrates SUMO with the OpenAI Gym environment, facilitating RL training in TSC.

Intelligent traffic control. In a multi-agent system, domain knowledge becomes a key for communication and coordination between agents. Some early methods such as PressLight [28] have achieved good single-agent control and proposed feasible training methods. MPLight [4] is based on PressLight and extends it to large road networks, using the same model to make decisions for all intersections, which requires that the state space and action space of each intersection are consistent. After Afshin Oroojlooy et al. [18] introduced the attention mechanism into this field, the GAT method gradually became mainstream. From a multi-agent perspective, several traffic lights are usually regarded as multiple different agents, and the road network is regarded as a topological structure to model the data structure. In this case, GAT becomes the main optimization method. For example, CoLight [29] is based on the DQN method and uses GAT to assign weights to neighbors. Experiments show that each agent works best when it only pays attention to itself and its one-hop neighbors. STMARL [26] and DynSTGAT [30] use a LSTM [8] or TCN [12] to capture historical state information (such as traffic flow) and use a graph convolutional network (GCN) or GAT to obtain spatial dependencies. DuaLight [15] introduces scene characteristics based on CoLight, introduces neighbor weighted matrices and feature-weighted matrices for each agent, and also performs GAT representation on the one-hop neighbors, further enhancing the agent's understanding of its own scene and local coordination capabilities. GuideLight [9] implements a control method that is closer to industrial needs based on cyclic phase switching and combined with behavioral cloning and curriculum learning training models.

3 NOTATION

We define the key concepts in RL for TSC before introducing our model, including the signal configuration and modeling TSC as a Partially Observable Markov Decision Process (POMDP).

Intersection. Fig. 1 shows a general right-hand 2-way 6-lane intersection. We define the traffic light numbers starting from the north and proceeding clockwise. The left-turn lanes can only be used for left turns, while the right-turn lanes can be used for both going straight and turning right. Among these, traffic lights numbered 1, 5, 9, and 13 are for right-turn lanes and are default to green. However, when going straight in a right-turn lane, one must obey

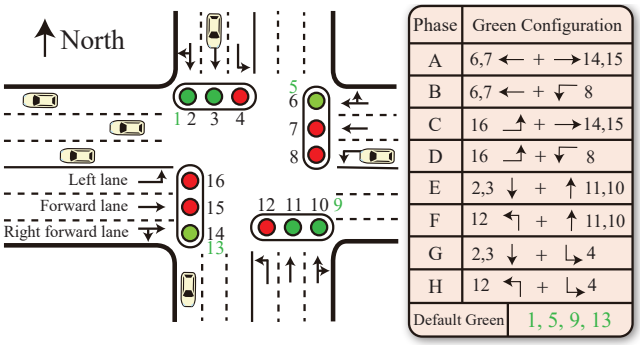


Figure 1: General right-hand 2-way 6-lane intersection with eight non-conflicting green light signal configurations. The right turn lane signal is green by default. Traffic signal numbers start from the north and go clockwise.

the green signal of going straight. Each time 3 or 4 green light signals are given to allow passing, a total of 8 signal combinations are defined in the scenario.

Phase. Referring to the table on the right side of Fig. 1, we define an equitable signal configuration scheme that ensures no lane conflicts exist for any passing scenarios within a single cycle and each lane has two opportunities for passing within the cycle. This scheme is consistent with most real-world configurations, and the algorithm can adapt individually even if there are different configuration schemes.

POMDP Modeling for TSC. The traffic signal control problem is modeled as a POMDP. We consider each intersection as an independent agent that faces continuously changing traffic conditions and can only observe its own information completely, without grasping the global state. Another principle is that the next state is only affected by the current state and the current decision, and has nothing to do with the previous state. A POMDP can be described by a tuple $\langle \mathcal{S}, \mathcal{O}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \pi, \gamma, \rangle$ and is introduced below.

Global state space $\mathcal{S}$ & Partial state space $\mathcal{O}$. The partial observation of agent i at time t is $o_i^t \in \mathcal{O}$, while global state $s^t \in \mathcal{S}$ and $o_i^t \in s^t$. Partial observations are also called local observations in following context. Refer to [2], each local observation consists of four parts:

1. The current action represented as a one-hot vector;
2. A boolean value indicating whether the current signal allows switching. We specify that each action must remain in place for at least 10 seconds to meet real-world requirements;
3. The vehicle density in each lane, calculated as the number of vehicles in the lane divided by the lane capacity;
4. The density of waiting vehicles in each lane, calculated as the number of stopped vehicles divided by the lane capacity;

These components are encoded into a vector to represent the current state of each intersection.

Action $\mathcal{A}$. In the case of Fig. 1, the eight phases correspond to eight different action choices. At time t, the action of agent i is $a_i^t \in \mathcal{A}$. In the simulation, by default, we provide the corresponding yellow signal before switching the red signal.

Table 1: Comparison of different reward methods. The first column includes various reward targets and a baseline, and the first row is the system indicators. Arrows indicate the better direction, and standard deviations in brackets are obtained from multiple experiments. The fixed time is similar to the real-life solution, that is, the fixed time switching signal, which is a baseline.

	Waiting↓	Queue↓	Speed↑
Pressure	2106.7 (1283)	40.8 (7)	7.9 (0.4)
Queue	4358.5 (2608)	50.4 (8)	7.5 (0.3)
Speed	1009.9 (597)	31.5 (9)	8.4 (0.4)
Waiting	790.1 (703)	23.8 (10)	8.8 (0.6)
Fixed time	684.8	70.2	8.1
Our adoption	**422.0 (577)**	**21.9 (11)**	**9.0 (0.6)**

Transition probability $\mathcal{P}$. Due to the Markov property, the probability transfer function is expressed as $\mathcal{P}(s^{t+1}|s^t, a^t)$. The specific form of the function is unknown and is usually represented by reality or a simulator. We perform RL to capture the dynamic characteristics.

Reward $\mathcal{R}$. Referring to the design of Alegre et al. [2], we first define the vehicle waiting time. At time t, the total waiting time of all vehicles stopped at intersection (agent) i is denoted as W_i^t. Then the reward of the agent is $r_i^t = W_i^{t-1} - W_i^t$. Our goal is to maximize the reward, which means that the agent should try to make the current waiting time shorter than the previous waiting time. The final reward is expected to converge to around 0, i.e., the system reaches a state of equilibrium. The advantage of considering waiting time as a reward is that the agent will not deliberately delay the release time of some lanes due to fewer cars there, but instead balanced take all vehicles into consideration.

There are many reward functions. For example, the reward value can increase with the decrease in the number of blocked vehicles, or set a pressure indicator [28] to measure the difference between the number of vehicles entering and leaving the lane. We test various reward functions in "ingolstadt21" [3], and this scenario is completely different from ours. We adopt the same algorithm independent PPO (IPPO) for all experiments. In this case, we evaluate various indicators and determine that the aforementioned method is the best choice, with superior performance compared to other methods. The experimental results are shown in Table 1.

Policy π. The decision made by agent i in time t based on the current partial observation o_i^t is given by policy function $\pi_i^t(a_i^t|o_i^t)$. The agent policy should maximize the total reward $\sum_{t=\tau}^{T} \gamma^{t-\tau} r_i^t$, where γ is the discount factor, usually 0.98. This means that agents discount future reward and care first about near-term reward.

4 METHODOLOGY

In this section, we will introduce the implementation of MacLight, including information aggregation, VAE feature compression, PPO, and method of dynamic traffic flow construction.

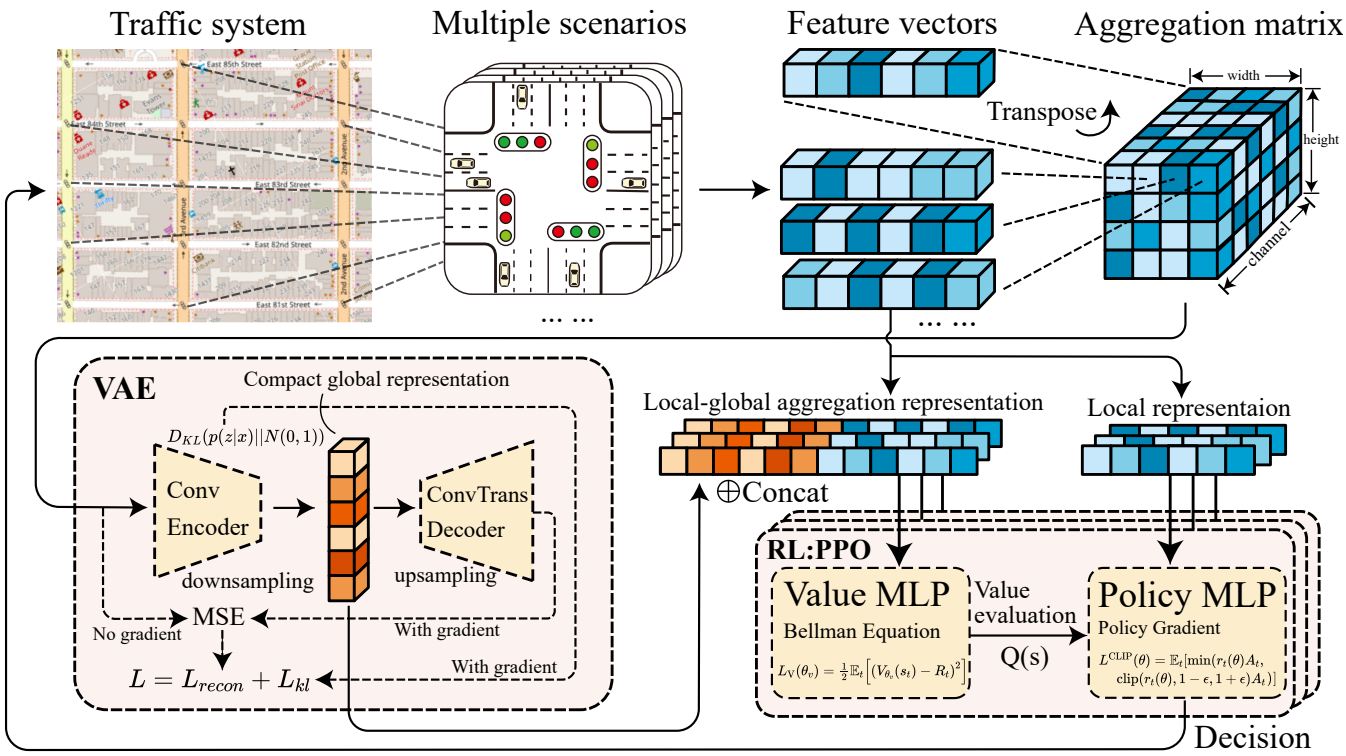


Figure 2: MacLight framework. The first row shows how to construct the aggregation matrix and the second row introduces the main model frameworks, including VAE and PPO.

4.1 Multi-scene aggregation matrix

Considering the geographical invariance of intersections, we organize the global information into a three-dimensional matrix. The global information is obtained by merging several local information, and each local information can be regarded as a scene. The local information of each intersection can be represented as a feature vector. Each vector is appropriately transposed and organized according to its location in geographic space, ultimately forming a high-dimensional global feature matrix as shown in the upper right of Fig. 2. The width and height of the matrix correspond to the geographical locations, and the number of channels is equal to the length of a single feature vector.

Clearly, the grid-based setting is the foundation for adopting CNN as the representation model. Although real-world road networks do not appear as regular as pixel grids, considering that most intersections are four-armed, the grid-like characteristics can still be observed when transforming them into a graph.

4.2 Autoencoder

For feature extraction of three-dimensional matrices, we construct a VAE based on CNN. The structure refers to the VAE in Fig. 2. The encoder performs downsampling and finally outputs a compact compressed representation. The decoder restores the representation to the original matrix. Its training is carried out according to the method of Diederik P. Kingma and Max Welling [10].

We first give the process of upsampling, for a matrix x with a channel length of 33, the process is

$$h = \text{Conv}_3^{256}\left[\text{ReLU}(\text{Conv}_3^{128}(\text{ReLU}(\text{Conv}_3^{64}(x))))\right]. \tag{1}$$

The parameters of the Gaussian distribution represented in the latent space are calculated as

$$\begin{aligned} \mu &= W_\mu h + b_\mu, \\ \log \sigma^2 &= W_{\text{logvar}} h + b_{\text{logvar}}, \end{aligned} \tag{2}$$

where W_μ, b_μ, W_{logvar} and b_{logvar} correspond to weights and biases respectively. Then, an effective compact representation z is obtained by Gaussian distributions built on μ and σ:

$$z = \mu + \epsilon \cdot \sigma, \quad \epsilon \sim \mathcal{N}(0, I). \tag{3}$$

The decoder uses transposed convolution models:

$$\begin{aligned} z_{reshape} &= \text{Reshape}(\text{Linear}(z)), \\ x_{\text{recon}} &= \text{Sigmoid}\Big[\text{ConvTrans}_3^{33}\Big[\text{ReLU}(\text{ConvTrans}_3^{64}(\\ &\qquad \text{ReLU}(\text{ConvTrans}_3^{128}(z_{reshape}))))\Big]\Big], \end{aligned} \tag{4}$$

where we use sigmoid activation for output because the value range of the observation vector is between 0 and 1. Thus, The loss function is expressed as:

$$L_{vae} = L_{recon} + L_{kl}, \tag{5}$$

where L_{recon} and L_{kl} are simply expressed as

$$L_{recon} = -\log p(x|z),$$
$$L_{kl} = -\frac{1}{2}\sum\left(1 + \log\sigma^2 - \mu^2 - \sigma^2\right). \tag{6}$$

In short, the VAE can be trained online during the RL training process. Due to the efficient calculation of CNN on GPU, the overall algorithm can maintain its advantage in saving time. The global feature representation z will be concatenated with the local feature to be local-global aggregation representations s_t^f and passed to the corresponding agent for PPO learning.

4.3 PPO

We adopt the PPO algorithm with Generalized Advantage Estimation (GAE) trick as backbone model, refer to bottom right of Fig. 2. The core idea of PPO is to update the policy by maximizing a clipped objective function, which helps prevent large updates that could destabilize training.

The policy function for PPO can be expressed as:

$$L^{CLIP}(\theta) = \mathbb{E}_t\left[\min\left(r_t(\theta)\hat{A}_t, \text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t\right)\right], \tag{7}$$

where $r_t(\theta)$ is the probability ratio defined as $\frac{\pi_\theta(a_t|o_t)}{\pi_{\theta_{\text{old}}}(a_t|o_t)}$. Here, π_θ denotes the policy parameterized by θ, a_t is the action taken, and o_t is the local observation at time t. The term $\hat{A}_t$ represents the estimated advantage, which quantifies how much better the taken action was compared to the expected action under the current policy.

We use GAE to compute the advantage estimate $\hat{A}_t$. It considers not only the immediate reward but also the value of future states, allowing for a more accurate approximation of advantage. The advantage can be computed as follows:

$$\hat{A}_t = \sum_{l=0}^{\infty}(\gamma\lambda)^l\delta_{t+l}, \tag{8}$$

where λ is a discount factor to balance short-term and long-term advantages, and δ_t is defined as:

$$\delta_t = r_t + \gamma V_\theta(s_{t+1}^f) - V_\theta(s_t^f), \tag{9}$$

where $V_\theta(s^f)$ represents the value function approximated by the neural network, r_t is the immediate reward, s_t^f is global-local aggregation representation introduced in the previous subsection, and γ is the discount factor that balances the importance of future rewards.

In addition, to updating the policy, the value function loss can be defined as:

$$L^V(\theta) = \mathbb{E}_t\left[(V_\theta(s_t^f) - V_{\text{target},t})^2\right], \tag{10}$$

where $V_{\text{target},t}$ is typically the sum of the immediate reward and the discounted value of the next state:

$$V_{\text{target},t} = r_t + \gamma V_\theta(s_{t+1}^f). \tag{11}$$

Through this structured approach, PPO with GAE provides a robust mechanism for policy updates while maintaining stability in learning, allowing for effective exploration and improved sample efficiency in the task. MacLight pseudocode is summarized in Algorithm 1. Ultimately, the algorithm will try to maximize the total reward to achieve the overall goal.

Algorithm 1 The pseudocode of MacLight

Ensure: The neural networks: $f^e, f^d, V^k; \pi^k$ // Encoder, Decoder, ValueNet, PolicyNet;
Initialize: L, T, K, E; // Training episodes, timesteps, number of intersections (agents), inner updating epoch of PPO;
for episode $l = 1$ to L **do**
  for timestep $t = 1$ to T **do**
    Global feature matrix s_t;
    Encoder global representation $s_t^g = f^e(s_t)$;
    Decoder reconstruction $s_t^r = f^d(s_t^g)$;
    Update the autoencoder f^e, f^d using Eq. 5;
    for agent $k = 1$ to K **do**
      Partial observation o_t^k, global representation s_t^g;
      Global-local representation $s_t^f = [s_t^g, o_t^k]$;
      Calculate advantage $\hat{A}_t$ using V, s_t^f by Eq. 8;
      for train epoch $e = 1$ to E **do**
        Update V^k using s_t^f by Eq. 10;
        Update π^k using o_t^k and $\hat{A}_t$ by Eq. 7;
      end for
    end for
  end for
end for

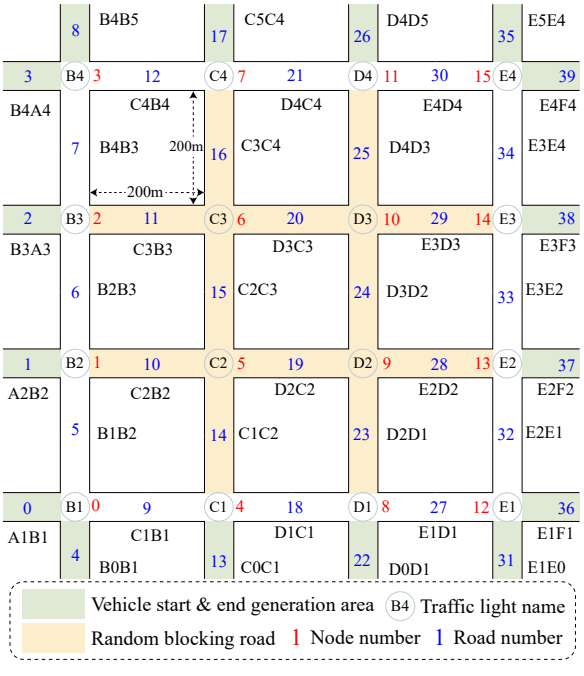


Figure 3: The road network of the simulation environment

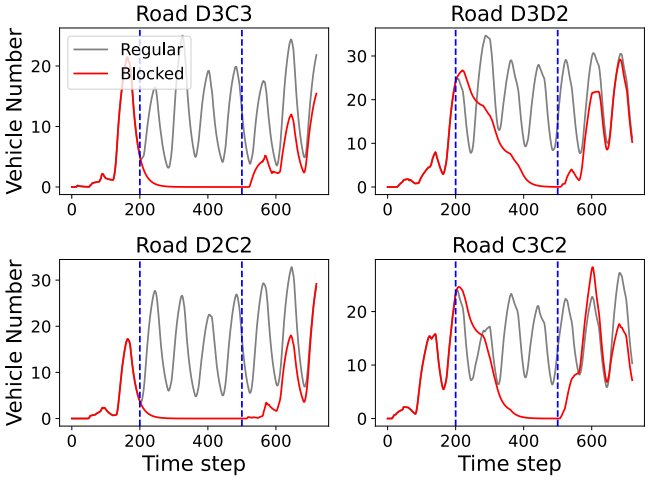


Figure 4: Some blocked lane traffic flow distributions

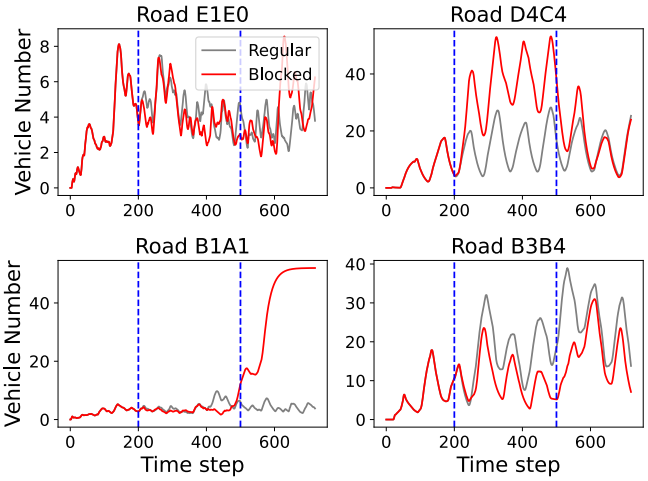


Figure 5: Some regular road traffic flow distributions

4.4 Dynamic traffic flow construction

In current TSC RL studies, the deployment of traffic flow is typically fixed, with all vehicles following predetermined routes. Our experiment, however, is the first to build a dynamic traffic flow environment that simulates emergency road events, which cause sudden changes in the distribution of traffic on other roads.

As illustrated in Fig. 3, when four central roads—D3C3, D3D2, D2C2, and C3C2—are blocked, all vehicles are asked to reroute. We conduct two experiments to analyze the effects. In Fig. 4, the gray curve shows the traffic flow distribution on a specific lane under normal conditions (without any interference), while the red curve shows the distribution after blocking a specific road. The lane blockage period is marked by a blue vertical dashed line. During this time, the traffic volume on the affected lanes drops significantly as vehicles select new optimal routes. Fig. 5 illustrates the changes in traffic flow on other, unblocked roads.

The impact of congestion on one road can be quite complex and influence other roads in unpredictable ways. Fig. 5 shows the flow distribution for unblocked lanes after the designated roads are closed. For example, while the traffic flow distribution on roads E1E10 remains largely unaffected, there is a significant increase in traffic on D4C4. On the other hand, B1A1 sees a sharp rise in traffic, while B3B4 experiences a decrease. Such interdependencies are difficult to model accurately but are common in real-world traffic systems. Consequently, our algorithm must account for these complex interactions.

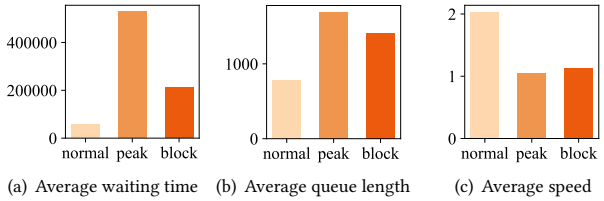


Figure 6: Statistics of different experimental scenarios

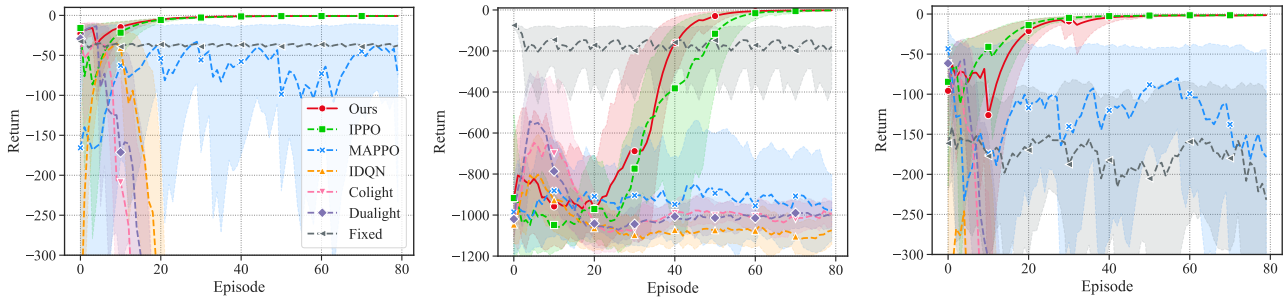
5 EXPERIMENTS

This section introduces the experimental environments, evaluation indicators, comparison algorithms, main experimental and ablation analysis. Table 3 can quickly check the experimental results.

5.1 Environment and metrics

Environment. In order to comprehensively evaluate algorithms, three different traffic scenarios are constructed on the same road network as shown in Fig. 3. The system is represented by a 4×4 grid arranged horizontally and vertically, with a distance of 200 meters. The three scenarios are a normal-pressure scenario with regular traffic flow called **Normal**, a high-pressure scenario with extremely high traffic flow called **Peak**, and a dynamic traffic scenario with normal traffic flow but random emergency road blockage called **Block**. The randomly blocked lanes are indicated by yellow parts in Fig. 3. In terms of task difficulty, the minimum traffic pressure for our scenarios is much greater than all the other current studies, refer to Table 2. Benchmarking our experimental scenarios, when all three scenarios adopt a fixed phase switching time of every 45 seconds, the system simulation statistics are shown in Fig. 6. All simulations are performed on the SUMO [13] simulation platform.

Metrics. In each scenario, we not only present the total reward results for all algorithms but also establish three objective metrics for comprehensive evaluation: the system's average waiting time, the queue length of waiting vehicles, and the average speed. These metrics take into account both temporal and spatial factors, enabling a more holistic assessment of the transportation system and preventing reward hacking [20]. The training seed range for all algorithms is set from 42 to 46, with details provided in the following subsection.



(a) Normal (b) Peak (c) Block

Figure 7: Training details of cumulative rewards

Table 2: Number of vehicles deployed on different scenario. Normal&Block and Peak are ours, while arterial4x4 and grid4x4 are the two similar scenarios tested in DuaLight.

	Normal&Block	Peak	arterial4x4	grid4x4
Vehicles	8000	10286	2485	1472

5.2 Comparison methods

MacLight will be compared with a variety of algorithms, including the traditional method of setting a fixed time switching phase and a variety of advanced algorithms based on RL.

The parameters of MacLight can be found in our code, where a clear table is also provided. The parameters of other methods are from the general settings or the corresponding papers and related codes.

Fixed time. Similar to the control method in reality, we configure the same fixed time switching method for all traffic lights: switching the phase every 45 seconds.

IPPO. Refer to [5]. A separate agent with PPO algorithm is constructed for each intersection, and each agent only focuses on its own local information. IPPO can be regarded as the ablation object of MacLight.

MAPPO. Refer to [31]. Similar to IPPO, but only one value evaluation network is used globally, whose input is the concatenation of local observations of all agents, while policy modules are as same as IPPO.

IDQN. Similar to IPPO, but replaces the PPO with DQN. IDQN is the backbone model of CoLight and DuaLight.

CoLight. Referring to [29], a strong algorithm for applying RL to TSC tasks using GAT, built on top of DQN.

DuaLight. Reference [15], a SOTA based on CoLight, adds feature weight matrix and neighborhood weight matrix for different scenarios to the backbone network for Q learning, which shows better representation effect than CoLight. It is also based on DQN.

5.3 Main results

Comparative experiments. Table 3 shows comprehensive comparison of experimental results. MacLight performs best in the Normal scene, with relatively good average performance and stability, followed by IPPO. In the high-pressure traffic environment Peak, the return and waiting time indicators are not as good as the Fixed method, because the indicators represent the average of the entire process, and if we check the final value, MacLight still has the best performance. In the dynamic traffic environment Block, indicators are inferior to IPPO. IDQN and the DQN-based CoLight and DuaLight methods perform poorly and are very prone to overfitting and policy collapse when faced with relatively sparse rewards and unstable data.

Training and testing. Fig. 7 shows the change of cumulative rewards during the entire training process, with the shadows indicating the maximum and minimum regions recorded for different seed experiments. On-policy approaches MacLight and IPPO, consistently demonstrates stable policy improvement across all scenarios. In contrast, off-policy methods such as IDQN and CoLight, while exhibiting robust initial performance, tend to collapse shortly thereafter. These methods are better suited for less challenging scenarios, leveraging the advantages of smaller models to avoid overfitting. However, they falter in high-difficulty, sparse-reward environments.

We show the test results of all algorithms in Table. 4, where the indicator is average return. These results show the scores achieved by each method with the best performance on a completely new seed environment. It can be noted that there are some differences with the training data, for example, MacLight is slightly better than IPPO in Block at this time, but lags behind IPPO in Normal. In short, although MacLight does not show a clear advantage over IPPO in some aspects, the cooperation mechanism is still an issue worth discussing.

Ablation analysis. IPPO in Table 3 can be regarded as an ablation experiment of MacLight, because MacLight modifies the input of the value module from local features to local-global aggregate representation. On most indicators, MacLight shows advantages, while the second-best method is IPPO.

Training time on wall clock. Table 5 shows the training time of MacLight compared to other off-policy algorithms. We tested multiple random number seeds, each seed trained 80 episodes, and each episode contained 3600 seconds simulation. The times in the table are calculated as the average of the total length of 80 episodes on each seed. MacLight requires less than 1 hour to train, while off-policy algorithm IDQN needs at least 2 hours, Colight and DuaLight are even slower. This is because the GCN-based method

Table 3: Experimental results of each scenario and indicator. IPPO can be considered as an ablation experiment. The specific values in the table include the mean of the current column indicator and the standard deviation in brackets. Best results in boldface, and the second-best results underlined. The preferred direction of the indicator is marked by up and down arrows. The waiting time is not given corresponding standard deviation due to the large value.

Scenario	Normal				Peak				Block			
Indicator	Return↑	Wait↓	Queue↓	Speed↑	Return↑	Wait↓	Queue↓	Speed↑	Return↑	Wait↓	Queue↓	Speed↑
Fixed time	-37.14(9)	56409	785(170)	2.1(1)	**-171.0**(98)	**292582**	1526(236)	1.2(1)	-179.5(59.3)	253944	1477(210)	1.1(0)
IPPO	-6.6(22)	10254	152(114)	6.2(1)	-434.8(451)	1258399	1456(1262)	2.8(3)	**-12.0**(28)	**13144**	**221**(197)	**5.4**(1)
MAPPO	-67.8(65)	122793	509(186)	3.3(1)	-924.2(117)	2559550	2942(165)	0.1(0)	-127.8(81)	190998	972(211)	1.7(0)
IDQN	-598.2(276)	1650798	2474(956)	0.7(1)	-1054.4(101)	4546469	3498(248)	0.0(0)	-796.6(198)	2625699	3136(610)	0.1(0)
CoLight	-716.2(283)	2538938	2913(969)	0.5(1)	-969.5(146)	4747313	3438(436)	0.0(0)	-788.1(228)	3360669	3186(772)	0.2(1)
DuaLight	-712.8(293)	2630974	2858(1009)	0.5(1)	-977.9(154)	4664564	3410(423)	0.0(0)	-770.5(246)	3221476	3146(816)	0.2(1)
MacLight	**-4.02**(10)	**4737**	**140**(90)	**6.3**(1)	-362.3(423)	998411	**1267**(1237)	**3.3**(3)	-17.3(44.0)	24224	249(237)	5.2(1)

Table 4: Test results of each algorithm on the average return. Slight differences from the training metrics can be noticed.

	Normal	Peak	Block
Fixed	-31	-100	-312
IPPO	**-0.68**	-1.60	-1.18
MAPPO	-202	-977	-206
IDQN	-842	-1051	-891
CoLight	-937	-997	-942
DuaLight	-878	-1034	-876
MacLight	-0.71	**-1.46**	**-1.17**

Table 5: Comparison of training wall time (minute) for 80 episodes between MacLight (Ours) and off-policy methods. All algorithms run on a single A100.

	Normal	Peak	Block
IDQN	137.4	178.9	186.4
CoLight	373.7	405.1	391.7
DuaLight	413.2	456.4	283.2
MacLight	43.0	58.1	39.1

cannot be massively parallelized, further slowing down the computational efficiency.

6 CONCLUSION

In this paper, we proposed the MacLight for TSC and construct both static and dynamic traffic flow for evaluation. The main contribution of MacLight is to construct a CNN-based VAE for global state feature extraction, and connect with the local state to form a local-global representation, which is used as the input of the value evaluation module to guide the policy improvement. MacLight uses the PPO algorithm as the backbone so that global and local information can be processed in parallel and improve each other. In addition, as an on-policy algorithm, MacLight provides high operating efficiency, taking only about one-third of the time of the off-policy method. Finally, the dynamic traffic simulation environment we constructed greatly expands the current research space and provides a basis for applying RL in emergency traffic scenarios.

7 LIMITATIONS

There is still room for improvement in our work. Although CNN is more efficient than GCN-based methods, real road networks are usually not as regular as Manhattan roads and cannot be directly constructed as pixel matrices. We can use multiscale convolution to alleviate this problem in the future. In addition, there is room for improvement in the reward function, such as introducing both local and global indicators.

ACKNOWLEDGMENTS

This work is supported by:

1. Hubei Provincial Key Laboratory of Metallurgical Industry Process System Science (Y202105)
2. High Performance Computing Center of Wuhan University of Science and Technology.

We also thank reviewers for their efforts and contributions.

REFERENCES

[1] Lucas N. Alegre. 2019. SUMO-RL. https://github.com/LucasAlegre/sumo-rl.
[2] Lucas N. Alegre, Ana L. C. Bazzan, and Bruno C. da Silva. 2021. Quantifying the impact of non-stationarity in reinforcement learning-based traffic signal control. *PeerJ Computer Science* 7 (2021), e575. https://doi.org/10.7717/peerj-cs.575
[3] James Ault and Guni Sharon. 2021. Reinforcement Learning Benchmarks for Traffic Signal Control. In *Proceedings of the Thirty-fifth Conference on Neural Information Processing Systems (NeurIPS 2021) Datasets and Benchmarks Track.*
[4] Chacha Chen, Hua Wei, Nan Xu, Guanjie Zheng, Ming Yang, Yuanhao Xiong, Kai Xu, and Zhenhui Li. 2020. Toward A Thousand Lights: Decentralized Deep Reinforcement Learning for Large-Scale Traffic Signal Control. *Proceedings of the AAAI Conference on Artificial Intelligence* 34, 04 (Apr. 2020), 3414–3421. https://doi.org/10.1609/aaai.v34i04.5744
[5] Christian Schroeder de Witt, Tarun Gupta, Denys Makoviichuk, Viktor Makoviychuk, Philip H. S. Torr, Mingfei Sun, and Shimon Whiteson. 2020. Is Independent Learning All You Need in the StarCraft Multi-Agent Challenge? arXiv:2011.09533 [cs.AI] https://arxiv.org/abs/2011.09533
[6] Christina Diakaki, Markos Papageorgiou, and Kostas Aboudolas. 2002. A multivariable regulator approach to traffic-responsive network-wide signal control. *Control Engineering Practice* 10, 2 (2002), 183–195. https://doi.org/10.1016/S0967-0661(01)00121-6
[7] Carlos Gershenson. 2005. Self-Organizing Traffic Lights. arXiv:nlin/0411066 [nlin.AO] https://arxiv.org/abs/nlin/0411066

[8] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-Term Memory. *Neural Comput.* 9, 8 (Nov. 1997), 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735

[9] Haoyuan Jiang, Xuantang Xiong, Ziyue Li, Hangyu Mao, Guanghu Sui, Jingqing Ruan, Yuheng Cheng, Hua Wei, Wolfgang Ketter, and Rui Zhao. 2024. GuideLight: "Industrial Solution" Guidance for More Practical Traffic Signal Control Agents. arXiv:2407.10811 [cs.MA] https://arxiv.org/abs/2407.10811

[10] Diederik P. Kingma and Max Welling. 2014. Auto-Encoding Variational Bayes. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, Yoshua Bengio and Yann LeCun (Eds.). http://arxiv.org/abs/1312.6114

[11] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. ImageNet Classification with Deep Convolutional Neural Networks. In *Advances in Neural Information Processing Systems*, F. Pereira, C.J. Burges, L. Bottou, and K.Q. Weinberger (Eds.), Vol. 25. Curran Associates, Inc.

[12] Colin Lea, René Vidal, Austin Reiter, and Gregory D. Hager. 2016. Temporal Convolutional Networks: A Unified Approach to Action Segmentation. *CoRR* abs/1608.08242 (2016). arXiv:1608.08242

[13] Pablo Alvarez Lopez, Michael Behrisch, Laura Bieker-Walz, Jakob Erdmann, Yun-Pang Flötteröd, Robert Hilbrich, Leonhard Lücken, Johannes Rummel, Peter Wagner, and Evamarie Wießner. 2018. Microscopic Traffic Simulation using SUMO, In The 21st IEEE International Conference on Intelligent Transportation Systems. *IEEE Intelligent Transportation Systems Conference (ITSC)*.

[14] Yican Lou, Jia Wu, and Yunchuan Ran. 2022. Meta-Reinforcement Learning for Multiple Traffic Signals Control. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management* (Atlanta, GA, USA) *(CIKM '22)*. Association for Computing Machinery, New York, NY, USA, 4264–4268. https://doi.org/10.1145/3511808.3557640

[15] Jiaming Lu, Jingqing Ruan, Haoyuan Jiang, Ziyue Li, Hangyu Mao, and Rui Zhao. 2024. DuaLight: Enhancing Traffic Signal Control by Leveraging Scenario-Specific and Scenario-Shared Knowledge. In *Proceedings of the 23rd International Conference on Autonomous Agents and Multiagent Systems* (Auckland, New Zealand) *(AAMAS '24)*. International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, 1283–1291.

[16] Jinming Ma and Feng Wu. 2020. Feudal Multi-Agent Deep Reinforcement Learning for Traffic Signal Control. In *Proceedings of the 19th International Conference on Autonomous Agents and MultiAgent Systems* (Auckland, New Zealand) *(AAMAS '20)*. International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, 816–824.

[17] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. 2013. Playing Atari with Deep Reinforcement Learning. arXiv:1312.5602 [cs.LG]

[18] Afshin Oroojlooy, Mohammadreza Nazari, Davood Hajinezhad, and Jorge Silva. 2020. AttendLight: universal attention-based reinforcement learning model for traffic signal control. In *Proceedings of the 34th International Conference on Neural Information Processing Systems* (Vancouver, BC, Canada) *(NIPS '20)*. Curran Associates Inc., Red Hook, NY, USA, Article 343, 12 pages.

[19] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal Policy Optimization Algorithms. arXiv:1707.06347 [cs.LG]

[20] Joar Skalse, Nikolaus H. R. Howe, Dmitrii Krasheninnikov, and David Krueger. 2024. Defining and characterizing reward hacking. In *Proceedings of the 36th International Conference on Neural Information Processing Systems* (New Orleans, LA, USA) *(NIPS '22)*. Curran Associates Inc., Red Hook, NY, USA, Article 687, 12 pages.

[21] Richard S. Sutton. 1988. Learning to predict by the methods of temporal differences. *Machine Learning* 3, 1 (01 Aug 1988), 9–44. https://doi.org/10.1007/BF00115009

[22] Richard S Sutton and Andrew G Barto. 2018. *Reinforcement learning: An introduction*. MIT press.

[23] Richard S. Sutton, David McAllester, Satinder Singh, and Yishay Mansour. 1999. Policy Gradient Methods for Reinforcement Learning with Function Approximation. In *Proceedings of the 12th International Conference on Neural Information Processing Systems* (Denver, CO) *(NIPS'99)*. MIT Press, Cambridge, MA, USA, 1057–1063.

[24] Pravin Varaiya. 2013. *The Max-Pressure Controller for Arbitrary Networks of Signalized Intersections*. Springer New York, New York, NY, 27–66. https://doi.org/10.1007/978-1-4614-6243-9_2

[25] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph Attention Networks. *International Conference on Learning Representations* (2018).

[26] Y. Wang, T. Xu, X. Niu, C. Tan, E. Chen, and H. Xiong. 2022. STMARL: A Spatio-Temporal Multi-Agent Reinforcement Learning Approach for Cooperative Traffic Light Control. *IEEE Transactions on Mobile Computing* 21, 06 (jun 2022), 2228–2242. https://doi.org/10.1109/TMC.2020.3033782

[27] Christopher J. C. H. Watkins and Peter Dayan. 1992. Q-learning. *Machine Learning* 8, 3 (01 May 1992), 279–292. https://doi.org/10.1007/BF00992698

[28] Hua Wei, Chacha Chen, Guanjie Zheng, Kan Wu, Vikash Gayah, Kai Xu, and Zhenhui Li. 2019. PressLight: Learning Max Pressure Control to Coordinate Traffic Signals in Arterial Network. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (Anchorage, AK, USA) *(KDD '19)*. Association for Computing Machinery, New York, NY, USA, 1290–1298. https://doi.org/10.1145/3292500.3330949

[29] Hua Wei, Nan Xu, Huichu Zhang, Guanjie Zheng, Xinshi Zang, Chacha Chen, Weinan Zhang, Yanmin Zhu, Kai Xu, and Zhenhui Li. 2019. CoLight: Learning Network-level Cooperation for Traffic Signal Control. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management* (Beijing, China) *(CIKM '19)*. Association for Computing Machinery, New York, NY, USA, 1913–1922. https://doi.org/10.1145/3357384.3357902

[30] Libing Wu, Min Wang, Dan Wu, and Jia Wu. 2021. DynSTGAT: Dynamic Spatial-Temporal Graph Attention Network for Traffic Signal Control. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management* (Virtual Event, Queensland, Australia) *(CIKM '21)*. Association for Computing Machinery, New York, NY, USA, 2150–2159. https://doi.org/10.1145/3459637.3482254

[31] Chao Yu, Akash Velu, Eugene Vinitsky, Jiaxuan Gao, Yu Wang, Alexandre Bayen, and Yi Wu. 2022. The Surprising Effectiveness of PPO in Cooperative Multi-Agent Games. In *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

[32] Dongbin Zhao, Yujie Dai, and Zhen Zhang. 2012. Computational Intelligence in Urban Traffic Signal Control: A Survey. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* 42, 4 (2012), 485–494. https://doi.org/10.1109/TSMCC.2011.2161577