MTLIGHT: EFFICIENT MULTI-TASK REINFORCEMENT LEARNING FOR TRAFFIC SIGNAL CONTROL

Liwen Zhu
School of Electronic and Computer Engineering, Shenzhen Graduate School
Peking University
Shenzhen, Guangdong, China
liwenzhu@pku.edu.cn

Peixi Peng, Zongqing Lu & Yonghong Tian
School of Computer Science
Peking University
Beijing, China
{pxpeng,zongqing.lu,yhtian}@pku.edu.cn

ABSTRACT

Traffic signal control has a great impact on alleviating traffic congestion in modern cities. Deep reinforcement learning (RL) has been widely used for this task in recent years, demonstrating promising performance but also facing many challenges such as limited performances and sample inefficiency. To handle these challenges, MTLIGHT is proposed to enhance the agent observation with a latent state, which is learned from numerous traffic indicators. Meanwhile, multiple auxiliary and supervisory tasks are constructed to learn the latent state, and two types of embedding latent features, the task-specific feature and task-shared feature, are used to make the latent state more abundant. Extensive experiments conducted on CityFlow demonstrate that MTLIGHT has leading convergence speed and asymptotic performance. We further simulate under peak-hour pattern in all scenarios with increasing control difficulty and the results indicate that MTLIGHT is highly adaptable.

1 INTRODUCTION

Traffic signal control aims to coordinate traffic signals across intersections to improve the traffic efficiency of a district or a city, which plays an important role in efficient transportation. Most conventional methods aim to control traffic signals by fixed-time Koonce & Rodegerdts (2008) or hand-crafted heuristics Kouvelas et al. (2014), which heavily rely on expert knowledge and in-depth excavation of regional historical traffic, making it difficult to migrate. Recently, deep reinforcement learning (DRL) based methods Guo et al. (2021); Jintao et al. (2020); Pan et al. (2020); He & Shin (2020); Tong et al. (2021); Wang et al. (2020); Gu et al. (2020); Liu et al. (2021); Xu et al. (2021); Zhang et al. (2021) employ a deep neural network to control an intersection where the network is learned by directly interacting with the environment. However, due to the plenty of traffic indicators (number of vehicles, queue length, waiting time, speed, etc.), complex observation and the dynamic environment, the problem is challenging and remains unsolved.

Since the observation, reward and dynamics of each traffic signal are closely related to others, hence optimizing traffic signal control in a large-scale road network is naturally modeled as a multi-agent reinforcement learning (MARL) problem. Most exiting works Wei et al. (2019a); Zhang et al. (2020b); Chen et al. (2020); Zheng et al. (2019a) are proposed to learn the policy of each agent only conditioned on the raw observations of the intersection, while ignoring the help of the global state, which is accessible in smart city. As stated in Zheng et al. (2019b), different metrics have a considerable impact on the traffic signal control task. Hence, the observation design of agent should not only involve the raw observations of the intersection, but also the global state. A good agent observation design could make full use of samples, and improves not only the policy performance but also the sample efficiency. However, there are a huge amount of traffic indicators or metrics in the global state, and it is hard to subjectively design suited and non-redundant agent observation among these indicators. On one hand, an overly concise observation design could not adequately and comprehensively represent the state characteristics and therefore affects the accuracy of the estimation of state transition and as well as influencing action selection. In contrast, if an overly complex combination of metrics is used as an observation, the weights of different metrics are difficult to precisely define, and it may cause data redundancy and dimension explosion, which will not only increase the computational consumption, but also make the agent hard to learn.

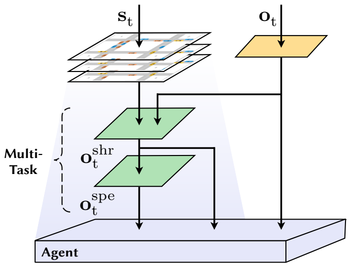


Figure 1: Multi-Task module forms task-shared and task-specific latent states to enhance the agent observation.

In order to provide an adequate representation of the traffic signal control task, the latent state is introduced. Specifically, the raw observation is identical to the intersection, which consists of several variables with concrete semantic meanings (i.e., the number of vehicles on each incoming lane and current signal phase). Then, the raw observation is enhanced by the latent space. To learn the latent space from the global state, multiple auxiliary and supervisory tasks are constructed, which are related to traffic signal control. That is, several statistics of global state history are taken as inputs, a RNN-based network is employed firstly, and several branches are introduced subsequently to predict multiple types of statistics of the global state, such as the flow distribution and the travel time distribution, respectively. To make the latent space more abundant, two types of embedding features are extracted: the task-specific feature and task-shared feature. The former is extracted by the task-specific branch and represents the task-driven information, while the later is from the task-shared layer and could express more general underlying characteristics. Hence, they are complementary to each other and are both used to enhance the raw observation. Finally, conditioned on the enhanced observation, the policy is learned by DRL Mnih et al. (2015). Note that the multiple tasks are learned simultaneously with the DRL, which makes the latent space more adaptive to the policy learning.

2 Problem Statement

2.1 Problem Definition

We consider a multi-agent traffic signal control problem, the task is modeled as a Markov Game Littman (1994), which can be denoted by a tuple $\mathcal{G} =< \mathcal{N}, \mathcal{S}, \mathcal{A}, \mathcal{O}, \mathcal{P}, \mathcal{R}, \mathcal{H}, \gamma >$. $\mathcal{N} \equiv \{1, \ldots, n\}$ is a finite set of agents, and each intersection in the scenario is controlled by an agent. $\mathcal{S}$ is a finite set of global state space. $\mathcal{A}$ denotes the action space for an individual agent. The joint action $\boldsymbol{a} \in \mathbf{A} \equiv \mathcal{A}^n$ is a collection of individual actions $[a_i]_{i=1}^n$. At each timestep, each agent i receives an observation $o_i \in \mathcal{O}$, selects an action a_i, results in the next state s' according to the transition function $\mathcal{P}(s' \mid s, \boldsymbol{a})$ and a reward $r = \mathcal{R}(s, \mathbf{a})$ for each agent. $\mathcal{H}$ is the time horizon and $\gamma \in [0, 1)$ is the discount factor.

2.2 Agent Design

Each intersection in the system is controlled by an agent. In the following, we introduce the state design, action design and reward design of the RL agent.

- **Observation.** Our primitive observation consists of two parts: (1) the number of vehicles on each incoming lane $\mathbf{f}_t^v$; (2) current signal phase $\mathbf{f}_t^s$. Both of them can be obtained directly from the simulator, the concepts are described in detail in Section B.4. The raw observation of agent i is defined by

$$o_i = \{\mathbf{f}_t^v, \mathbf{f}_t^s\}, \tag{1}$$

where $\mathbf{f}_t^v = \{V_{l_1^{in}}, V_{l_2^{in}}, \ldots, V_{l_m^{in}}\}$ and $l^{in} = \{l_1^{in}, \ldots, l_m^{in}\}$ is a finite set of incoming lanes in the intersection. Current signal phase $\mathbf{f}_t^s = p_k, k \in 1, \ldots, K$, and K is the total number of phases. Each phase p is represented as a one-hot vector. Our goal is to learn latent space to enhance the raw observation to make better use of the sample.

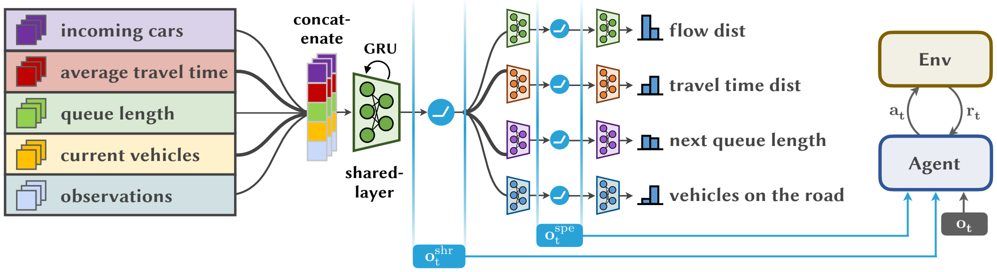


Figure 2: MTLight consists of a multi-task network and a policy network. RL agent is augmented with a task-shared latent state $\mathbf{o}_t^{\mathrm{shr}}$ and a task-specific latent state $\mathbf{o}_t^{\mathrm{spe}}$.

- **Action.** The action of each agent is to choose the phase for the next time interval. Note that the phases may organize in a sequential way in reality, while directly selecting a phase makes the traffic control plan more flexible. Action of agent i is defined by

$$a_i = \{\mathbf{f}_t^s\}, \tag{2}$$

  where $\mathbf{f}_t^s = p_k, k \in 1, \ldots, K$.

- **Reward.** We define the reward as the negative of the queue length on incoming lanes, which is generally accepted and reasonable in previous work Zheng et al. (2019b); Huang et al. (2021); Zang et al. (2020); Zheng et al. (2019a); Wei et al. (2019b). Reward of agent i is defined by

$$r_i = -\sum_{m}^{M} q_{l_m^{in}}, \tag{3}$$

  where $q_{l_m^{in}}$ is the queue length on incoming lane l_m^{in}.

3 Method

In this section, we will introduce the main modules of our proposed method MTLight, which focuses on learning task-related task-shared latent state and task-specific latent state by introducing an auxiliary Multi-Task network to help policy learning. The whole process of MTLight is described in Algorithm 1, and the framework of MTLight is shown in Fig. 2.

MTLight consists of a Multi-Task network and an agent network. For the latter, Deep Q-Network (DQN) Mnih et al. (2015) is employed as function approximator to estimate the Q-value function, which is consistent with the previous methods Chen et al. (2020); Wei et al. (2019b;a); Zheng et al. (2019a); Wei et al. (2018). The Multi-Task module adopts a hard parameter sharing paradigm Caruana (1997), which generally applied by sharing the hidden layers between all tasks, while keeping several task-specific output layers.

3.1 Multi-Task Learning for Latent State

For each agent, its raw observation includes the number of vehicles $\mathbf{f}_t^v$ and the current signal phase $\mathbf{f}_t^s$. Besides, several information from the global state is given, such as: the number of incoming cars in the last τ steps, denoted as $\mathbf{f}_{t-\tau:t}^c = [\mathbf{f}_{t-\tau}^c, \mathbf{f}_{t-\tau+1}^c, \ldots, \mathbf{f}_t^c]$, the average travel time during the past τ steps, denoted as $\mathbf{f}_{t-\tau:t}^{tr} = [\mathbf{f}_{t-\tau}^{tr}, \mathbf{f}_{t-\tau+1}^{tr}, \ldots, \mathbf{f}_t^{tr}]$, the queue length during the past τ steps, denoted as $\mathbf{f}_{t-\tau:t}^q = [\mathbf{f}_{t-\tau}^q, \mathbf{f}_{t-\tau+1}^q, \ldots, \mathbf{f}_t^q]$, the current vehicles during the past τ steps, which is denoted as $\mathbf{f}_{t-\tau:t}^{vr} = [\mathbf{f}_{t-\tau}^{vr}, \mathbf{f}_{t-\tau+1}^{vr}, \ldots, \mathbf{f}_t^{vr}]$.

The Multi-Task module includes the following four tasks:

1. **Flow distribution approximation.** We use $\mathcal{T}_{flow}$ to denote the traffic distribution estimation task, i.e., to predict the mean μ_f and variance σ_f^2 of flow arrival rate from start up to the time step t. The task could be denoted as:

$$(\mu_f, \sigma_f^2) \leftarrow [\mathbf{f}_t^v, \mathbf{f}_t^s, \mathbf{f}_{t-\tau:t}^c, \mathbf{f}_{t-\tau:t}^{tr}, \mathbf{f}_{t-\tau:t}^q, \mathbf{f}_{t-\tau:t}^{vr}]. \tag{4}$$

2. **Travel time distribution approximation.** We use $\mathcal{T}_{travel}$ to denote the travel distribution estimation task, i.e., to predict the mean μ_{tr} and variance σ^2_{tr} of average travel time of vehicles that have completed the trip from start up to the time step t:
$$(\mu_{tr}, \sigma^2_{tr}) \leftarrow [\mathbf{f}^v_t, \mathbf{f}^s_t, \mathbf{f}^c_{t-\tau:t}, \mathbf{f}^{tr}_{t-\tau:t}, \mathbf{f}^q_{t-\tau:t}, \mathbf{f}^{vr}_{t-\tau:t}]. \tag{5}$$
3. **Next queue length approximation.** We use $\mathcal{T}_{queue}$ to denote the next queue length estimation task, i.e., to predict the average number q of vehicles in queue at the next step:
$$q \leftarrow [\mathbf{f}^v_t, \mathbf{f}^s_t, \mathbf{f}^c_{t-\tau:t}, \mathbf{f}^{tr}_{t-\tau:t}, \mathbf{f}^q_{t-\tau:t}, \mathbf{f}^{vr}_{t-\tau:t}]. \tag{6}$$
4. **Vehicles on the road approximation.** We use $\mathcal{T}_{vehicles}$ to denote the vehicles on the road approximation task, i.e., to predict the number of vehicles V^r existing in the system:
$$V^r \leftarrow [\mathbf{f}^v_t, \mathbf{f}^s_t, \mathbf{f}^c_{t-\tau:t}, \mathbf{f}^{tr}_{t-\tau:t}, \mathbf{f}^q_{t-\tau:t}, \mathbf{f}^{vr}_{t-\tau:t}]. \tag{7}$$
Note that vehicles that have completed the trips or have not yet entered the road network do not belong to these.

The above tasks act auxiliary tasks to learn the latent space. Since the numbers of $\mathbf{f}^c_{t-\tau:t}$, $\mathbf{f}^{tr}_{t-\tau:t}$, $\mathbf{f}^q_{t-\tau:t}$, $\mathbf{f}^{vr}_{t-\tau:t}$ have different scales and their dimensions are different with $\mathbf{f}^v_t$ and $\mathbf{f}^s_t$, four independent linear layers and ReLU functions are employed firstly to scale them respectively:
$$\mathbf{h}^c = ReLU(\mathbf{W}_1 \mathbf{f}^c_{t-\tau:t} + \mathbf{b}_1),\ \mathbf{h}^{tr} = ReLU(\mathbf{W}_2 \mathbf{f}^{tr}_{t-\tau:t} + \mathbf{b}_2), \tag{8}$$
$$\mathbf{h}^q = ReLU(\mathbf{W}_3 \mathbf{f}^q_{t-\tau:t} + \mathbf{b}_3),\ \mathbf{h}^{vr} = ReLU(\mathbf{W}_4 \mathbf{f}^{vr}_{t-\tau:t} + \mathbf{b}_4). \tag{9}$$

Then a linear layer and ReLU function is used to calculate the hidden state after concatenating all embedded inputs:
$$\mathbf{H}_t = ReLU(\mathbf{W}(\mathbf{f}^v_t, \mathbf{f}^s_t, \mathbf{h}^c, \mathbf{h}^{tr}, \mathbf{h}^q, \mathbf{h}^{vr}) + \mathbf{b}). \tag{10}$$
Based on $\mathbf{H}_t$, a task-shared network module is used to generate its task-shared latent feature ($\mathbf{o}^{\mathrm{shr}}_{\mathrm{t}}$, also called *apparent state*). Then, 4 independent branches are introduced for each task and calculate task-specific latent feature ($\mathbf{o}^{\mathrm{spe}}_{\mathrm{t}}$, also called *mental state*) from $\mathbf{o}^{\mathrm{shr}}_{\mathrm{t}}$. The specific implementation of network architecture is listed in the supplementary.

We use a single latent variable model to extract hierarchical latent features, which follows insights by Zhao et al. (2017). That is, the *mental state* is output of the shared-layer after GRU in Multi-Task network and could express more general underlying characteristics. In contrast, the *apparent state* is the the concatenation of the output of the task-specific layer and represents the task-driven information. In other words, the *mental state* is more coarse-grained, while *apparent state* is more fine-grained. Hence, they are complementary to each other and both used in our method.

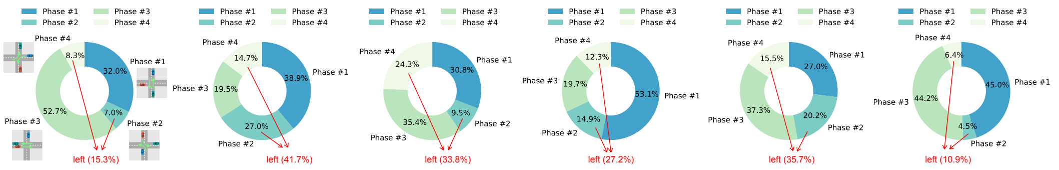
3.2 Policy with Latent State

With the help of latent state, the agent observation is enhanced from $\mathbf{o}_{\mathrm{t}}$ to $(\mathbf{o}_{\mathrm{t}}, \mathbf{o}^{\mathrm{shr}}_{\mathrm{t}}, \mathbf{o}^{\mathrm{spe}}_{\mathrm{t}})$. For the policy π^θ, the objective is to maximize the cumulative reward:
$$\max_\theta J(\theta) = \mathbb{E}_{a_t \sim \pi^\theta(a_t|\mathbf{o}_{\mathrm{t}}, \mathbf{o}^{\mathrm{shr}}_{\mathrm{t}}. \mathbf{o}^{\mathrm{spe}}_{\mathrm{t}})} \sum_{t=0}^{\mathcal{H}-1} \gamma^t r_{t+1}. \tag{11}$$

An agent that maximises Eq. 11 acts optimally under uncertainty and is called *Bayes-optimal* Ghavamzadeh et al. (2015), assuming we treat the knowledge over related tasks as our epistemic prior about the environment. Multi-Task module minimizes the complexity of the model and give informative priors to the model. Besides, it can minimize the representation bias in a way that push the learning algorithm to find a solution on a smaller area of representations on the intersection rather than on a large area of a single task. This incentivises a faster and better convergence.

4 Experiment

We conduct the experiments on CityFlow Zhang et al. (2019), an city-level open-source simulation platform for traffic signal control. The simulator is used as the environment to provide state for traffic



(a) MTLight (b) Individual RL (c) MetaLight (d) PressLight (e) CoLight (f) GeneraLight

Figure 3: Illustration of strategies for all RL methods under *Real* configuration in Hangzhou.

signal control, the agents execute actions by changing the phase of traffic lights, and the simulator returns feedback.

Please refer to Appendix D.1 and Appendix D.2 for the detailed settings of road network and traffic flow configuration. Baselines are described in detail in Appendix F.

4.1 Performance Comparison

Table 1: Overall performance comparison on Hangzhou, Jinan, New York and Shenzhen under *Real* and *Synthetic* configurations. Average travel time is reported in the unit of second. "Mean" in the last column shows the average performance of the scenarios shown in the previous 8 columns.

Model	**Hangzhou**		**Jinan**		**Newyork**		**Shenzhen**		**Mean**
	real	**syn_peak**	**real**	**syn_peak**	**real**	**syn_peak**	**real**	**syn_peak**	
MaxPressure	416.82	2320.65	355.12	1218.13	380.42	1481.48	**389.45**	1387.87	1387.87
FixedTime	718.29	1787.58	814.09	1739.69	1849.78	2086.59	786.54	1845.03	1453.45
SOTL	1209.26	2062.49	1453.97	1991.03	1890.55	2140.15	1376.52	2098.09	1777.76
Individual RL	743.00	1819.57	843.63	1745.07	1867.86	2100.68	769.47	1845.34	1466.83
MetaLight	480.77	1576.32	784.98	1854.38	261.34	2145.49	694.83	2083.26	1235.17
PressLight	529.64	1754.09	809.87	1930.98	302.87	1846.76	639.04	1832.76	1205.75
CoLight	297.89	1077.29	511.43	1217.17	**159.81**	1457.56	438.45	1367.38	815.87
GeneraLight	335.18	1574.93	585.89	1616.28	1208.73	1686.49	792.22	1574.10	1171.73
Base	705.85	1718.37	808.28	1703.21	903.82	2097.84	728.49	1937.45	1325.41
Base+raw	684.34	1845.92	623.94	1835.45	592.34	1934.04	703.56	1845.32	1258.11
Base+shr	313.28	1146.79	499.88	1325.27	463.15	1416.65	438.69	1371.53	871.91
Base+spe	431.55	1446.63	517.09	1430.96	431.65	1669.61	684.83	1442.35	1006.83
MTLight	**161.24**	**1011.67**	**346.93**	**1176.02**	209.46	**1394.15**	402.57	**1284.93**	**748.37**

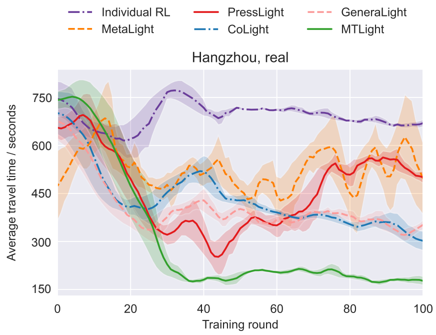


Figure 4: Performance of RL methods under real configurations.

Tab. 1 lists the comparative results, and it is evident that: 1) In general, RL methods perform better than conventional methods, and it indicates the advantage of the RL. Moreover, MTLight is outperforms other methods in almost all cities and flow configurations, which demonstrates the effectiveness of the method. 2) MTLight shows good generalization for different scenarios and configurations. For example, MaxPressure performs well in $\mathcal{D}_{Hangzhou}$ with the *Real*, while under the *Synthetic* traffic conditions, MaxPressure shows significantly worse than other methods. In contrast, MTLight can not only achieve good performance under diverse configurations of $\mathcal{D}_{Hangzhou}$, but also shows great stability. 3) MTLight outperforms Individual RL, MetaLight and PressLight with 693.46, 461.80 and 432.38, respectively. The reason is that they learn the traffic light's policy only using its observation and ignore the influence of the neighbors, while MTLight considers the neighbors as the latent part of the environment to help learning. 4) The neighbor's information is modeled in CoLight and GeneraLight can adapt to a variety of flows, they both perform well. While results of MTLight is superiors to them in multiple scenarios, resulting mean 42.5 and 398 improvement. Compared to them, MTLight benefits from prior knowledge learned from Multi-Task network to make more accurate decisions.

Fig. 4 shows the performances of all RL methods of $\mathcal{D}_{Hangzhou}$ under *Real* traffic pattern, and it is obvious that MTLight converges faster and has better asymptotic performance. Fig. 5 shows the performances of all RL methods of $\mathcal{D}_{Hangzhou}$ under *Synthetic* traffic pattern, we can conclude that MTLight converges quickly and learns effectively during the peak hour, while the other method have only a weak boost during training.

Fig. 8 and Tab. 5 illustrates the turning statistics of vehicle routes. Take $\mathcal{D}_{Hangzhou}$ *Real* as an example, the frequency of turning left and going straight is 14% and 86% respectively (turning right are not considered because they are free from the control by lights). Fig. 3 shows the percentage of each phase of RL methods, we can find: 1) The total left-turn phase of MTLight accounts for 15.3%, which is highly consistent with the left-turn frequency of 14%, which indicates that the strategy is interpretable. 2) The GeneraLight left-turn ratio of 10.9% is also close, but because it has an excessive proportion of straight phases, it may cause left-turn vehicles to be stranded, resulting in increased travel time. 3) Individual RL tends to consider phase 1 and 2, which account for as much as 65.9%, MetaLight prefers to go straight, PressLight is eccentric to phase 1, and CoLight assigns a relatively even distribution to each phase, rather than aligning with the traffic flow direction. These all demonstrate the limitations of other RL methods in multi-agent environments, while MTLight can learn more stable strategies by introducing task-shared and task-specific latent states.

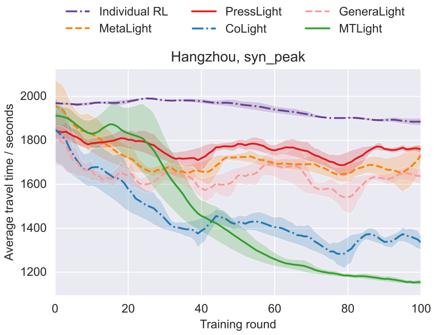


Figure 5: Performance of RL methods under synthetic peak configurations.

4.2 Ablations

To better validate the contribution of each component, three variants of MTLight are evaluated under a variety of scenarios, as shown in Tab. 1.

- **Base** only keeps the policy network and removes the Multi-Task network.
- **Base+raw** only keeps the policy network and discards Multi-Task network, but directly uses the original input of Multi-Task module as part of the observation.
- **Base+shr** retains the Multi-Task network and the policy, but only has task-shared latent state and removes task-specific latent state.
- **Base+spe** retains the Multi-Task network and the policy. In contrast to Base+shr, Base+spe has only the task-specific latent state and removes the task-shared latent state.

Note that MTLight contains the whole modules: policy network, Multi-Task network with both task-specific latent state and task-shared latent state.

The quantitative evaluation results are presented in Tab. 1. We can obtain the following findings: 1) Among these 4 models, the performance of Base is the worst. The reason is that it is hard to learn the effective policy independently in the multi-agent traffic signal control task, where the surrounding environment is changing dynamically, but Base has no sense of it. 2) Compared with the Base and Base+raw, the improvement of Base+shr and Base+spe demonstrate the effectiveness of the task-shared latent state $\mathbf{o}_{\mathrm{t}}^{\mathrm{shr}}$ and task-specific latent state $\mathbf{o}_{\mathrm{t}}^{\mathrm{spe}}$ respectively. $\mathbf{o}_{\mathrm{t}}^{\mathrm{shr}}$ reflects prior information that is constant over time with multiple related tasks , $\mathbf{o}_{\mathrm{t}}^{\mathrm{spe}}$ reflects prior information that is align with the latest changing trends, both of them help policy to make Bayesian optimal decisions. 3) The $\mathbf{o}_{\mathrm{t}}^{\mathrm{shr}}$ and $\mathbf{o}_{\mathrm{t}}^{\mathrm{spe}}$ are both effective because each of them is an efficient representations of environmental features. Compared to them, the superiority of MTLight indicates $\mathbf{o}_{\mathrm{t}}^{\mathrm{shr}}$ and $\mathbf{o}_{\mathrm{t}}^{\mathrm{spe}}$ are complementary to each other. Overall, all of the proposed components contribute positively to the final results.

5 CONCLUSION

We introduced MTLIGHT, an efficient Multi-Task reinforcement learning method for traffic signal control that can be scaled to complex multi-agent urban road networks of different scale. We showed that MTLIGHT's latent structure learns a hierarchical latent representations of related tasks, separating the task-shared and task-specific latent states. On several cities' datasets we demonstrated that this latent representation inspired from related multiple tasks, and conditioning the policy on it, allows an agent to adapt to the complex environment. We conclude that maintaining prior approximations over related tasks helps compared to model-free approaches, especially when there is too much information in the environment and it cannot be fully expressed by artificial state design.

For the future, the latent prior could be learned from expert data prepared in advance using imitation learning techniques Song et al. (2018), or by using existing multi-agent algorithms to pre-train Multi-Task network.

REFERENCES

Monireh Abdoos, Nasser Mozayani, and Ana LC Bazzan. Traffic light control in non-stationary environments based on multi agent q-learning. In *ITSC*. IEEE, 2011.

Monireh Abdoos, Nasser Mozayani, and Ana LC Bazzan. Holonic multi-agent system for traffic signals control. *Engineering Applications of Artificial Intelligence*, 2013.

Itamar Arel, Cong Liu, Tom Urbanik, and Airton G Kohls. Reinforcement learning-based multi-agent system for network traffic signal control. *IET Intelligent Transport Systems*, 2010.

Marc Bellemare, Will Dabney, Robert Dadashi, Adrien Ali Taiga, Pablo Samuel Castro, Nicolas Le Roux, Dale Schuurmans, Tor Lattimore, and Clare Lyle. A geometric perspective on optimal representations for reinforcement learning. *Advances in neural information processing systems*, 32, 2019.

Rich Caruana. Multitask learning. *Machine learning*, 1997.

Chacha Chen, Hua Wei, Nan Xu, Guanjie Zheng, Ming Yang, Yuanhao Xiong, Kai Xu, and Zhenhui Li. Toward a thousand lights: Decentralized deep reinforcement learning for large-scale traffic signal control. In *AAAI*, 2020.

Stephen Chiu. Adaptive traffic signal control using fuzzy logic. In *Proceedings of the Intelligent Vehicles92 Symposium*. IEEE, 1992.

Stephen Chiu and Sujeet Chand. Self-organizing traffic control via fuzzy logic. In *IEEE Conference on Decision and Control*. IEEE, 1993.

Tianshu Chu, Jie Wang, Lara Codecà, and Zhaojian Li. Multi-agent deep reinforcement learning for large-scale traffic signal control. *ITS*, 2019.

Seung-Bae Cools, Carlos Gershenson, and Bart D'Hooghe. Self-organizing traffic lights: A realistic simulation. In *Advances in applied self-organizing systems*. Springer, 2013.

Ivana Dusparic and Vinny Cahill. Distributed w-learning: Multi-policy optimization in self-organizing systems. In *self-adaptive and self-organizing systems*. IEEE, 2009.

Samah El-Tantawy, Baher Abdulhai, and Hossam Abdelgawad. Multiagent reinforcement learning for integrated network of adaptive traffic signal controllers (marlin-atsc): methodology and large-scale application on downtown toronto. *IEEE TITS*, 2013.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *ICML*. PMLR, 2017.

Mohammad Ghavamzadeh, Shie Mannor, Joelle Pineau, Aviv Tamar, et al. Bayesian reinforcement learning: A survey. *Foundations and Trends® in Machine Learning*, 2015.

Jingjing Gu, Qiang Zhou, Jingyuan Yang, Yanchi Liu, Fuzhen Zhuang, Yanchao Zhao, and Hui Xiong. Exploiting interpretable patterns for flow prediction in dockless bike sharing systems. *IEEE Transactions on Knowledge and Data Engineering*, 2020.

Xin Guo, Zhengxu Yu, Pengfei Wang, Zhongming Jin, Jianqiang Huang, Deng Cai, Xiaofei He, and Xiansheng Hua. Urban traffic light control via active multi-agent communication and supply-demand modeling. *IEEE Transactions on Knowledge and Data Engineering*, 2021.

Suining He and Kang G Shin. Spatio-temporal capsule-based reinforcement learning for mobility-on-demand coordination. *IEEE Transactions on Knowledge and Data Engineering*, 2020.

Xingshuai Huang, Di Wu, Michael Jenkin, and Benoit Boulet. Modellight: Model-based meta-reinforcement learning for traffic signal control. *arXiv preprint arXiv:2111.08067*, 2021.

PB Hunt, DI Robertson, RD Bretherton, and RI Winton. Scoot-a traffic responsive method of coordinating signals. Technical report, 1981.

Max Jaderberg, Volodymyr Mnih, Wojciech Marian Czarnecki, Tom Schaul, Joel Z Leibo, David Silver, and Koray Kavukcuoglu. Reinforcement learning with unsupervised auxiliary tasks. *arXiv preprint arXiv:1611.05397*, 2016.

Qize Jiang, Jingze Li, Weiwei Sun SUN, and Baihua Zheng. Dynamic lane traffic signal control with group attention and multi-timescale reinforcement learning. IJCAI, 2021.

KE Jintao, Hai Yang, Jieping Ye, et al. Learning to delay in ride-sourcing systems: a multi-agent deep reinforcement learning framework. *IEEE Transactions on Knowledge and Data Engineering*, 2020.

Peter Koonce and Lee Rodegerdts. Traffic signal timing manual. Technical report, United States. Federal Highway Administration, 2008.

Anastasios Kouvelas, Jennie Lioris, S Alireza Fayazi, and Pravin Varaiya. Maximum pressure controller for stabilizing queues in signalized arterial networks. *Transportation Research Record*, 2014.

Lior Kuyer, Shimon Whiteson, Bram Bakker, and Nikos Vlassis. Multiagent reinforcement learning for urban traffic control using coordination graphs. In *ECML-PKDD*. Springer, 2008.

Xingyu Lin, Harjatin Baweja, George Kantor, and David Held. Adaptive auxiliary task weighting for reinforcement learning. *Advances in neural information processing systems*, 2019.

Michael L Littman. Markov games as a framework for multi-agent reinforcement learning. In *Machine learning proceedings*. Elsevier, 1994.

Jia Liu, Tianrui Li, Shenggong Ji, Peng Xie, Shengdong Du, Fei Teng, and Junbo Zhang. Urban flow pattern mining based on multi-source heterogeneous data fusion and knowledge graph embedding. *IEEE Transactions on Knowledge and Data Engineering*, 2021.

PR Lowrie. Scats, sydney co-ordinated adaptive traffic system: A traffic responsive method of controlling urban traffic. 1990.

Clare Lyle, Mark Rowland, Georg Ostrovski, and Will Dabney. On the effect of auxiliary tasks on representation dynamics. In *International Conference on Artificial Intelligence and Statistics*. PMLR, 2021.

Patrick Mannion, Jim Duggan, and Enda Howley. An experimental review of reinforcement learning algorithms for adaptive traffic signal control. In *Autonomic road transport support systems*. Springer, 2016.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 2015.

Anthony Ndirango and Tyler Lee. Generalization in multitask deep neural classifiers: a statistical physics approach. *Advances in Neural Information Processing Systems*, 2019.

Tomoki Nishi, Keisuke Otaki, Keiichiro Hayakawa, and Takayoshi Yoshimura. Traffic signal control based on reinforcement learning with graph convolutional neural nets. In *ITSC*. IEEE, 2018.

Junhyuk Oh, Satinder Singh, Honglak Lee, and Pushmeet Kohli. Zero-shot task generalization with multi-task deep reinforcement learning. In *ICML*. PMLR, 2017.

Afshin Oroojlooy, Mohammadreza Nazari, Davood Hajinezhad, and Jorge Silva. Attendlight: Universal attention-based reinforcement learning model for traffic signal control. *arXiv preprint arXiv:2010.05772*, 2020.

Zheyi Pan, Wentao Zhang, Yuxuan Liang, Weinan Zhang, Yong Yu, Junbo Zhang, and Yu Zheng. Spatio-temporal meta learning for urban traffic prediction. *IEEE Transactions on Knowledge and Data Engineering*, 2020.

Stefano Giovanni Rizzo, Giovanna Vantini, and Sanjay Chawla. Time critic policy gradient methods for traffic signal control in complex and congested scenarios. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2019.

Roger P Roess, Elena S Prassas, and William R McShane. *Traffic engineering*. Pearson/Prentice Hall, 2004.

Sebastian Ruder. An overview of multi-task learning in deep neural networks. *arXiv preprint arXiv:1706.05098*, 2017.

Jiaming Song, Hongyu Ren, Dorsa Sadigh, and Stefano Ermon. Multi-agent generative adversarial imitation learning. *Advances in neural information processing systems*, 2018.

Torgny Svanes and James R Delaney. Scat: System control analysis and training simulator. In *Human Detection and Diagnosis of System Failures*. Springer, 1981.

Yongxin Tong, Dingyuan Shi, Yi Xu, Weifeng Lv, Zhiwei Qin, and Xiaocheng Tang. Combinatorial optimization meets reinforcement learning: Effective taxi order dispatching at large-scale. *IEEE Transactions on Knowledge and Data Engineering*, 2021.

T Tongloy, S Chuwongin, K Jaksukam, C Chousangsuntorn, and S Boonsang. Asynchronous deep reinforcement learning for the mobile robot navigation with supervised auxiliary tasks. In *International Conference on Robotics and Automation Engineering (ICRAE)*, pp. 68–72. IEEE, 2017.

Elise Van der Pol and Frans A Oliehoek. Coordinated deep reinforcement learners for traffic light control. *NeurIPS*, 2016.

Pravin Varaiya. The max-pressure controller for arbitrary networks of signalized intersections. In *Advances in Dynamic Network Modeling in Complex Transportation Systems*. Springer, 2013.

Senzhang Wang, Jiannong Cao, and Philip Yu. Deep learning for spatio-temporal data mining: A survey. *IEEE transactions on knowledge and data engineering*, 2020.

Fo Vo Webster. Traffic signal settings. Technical report, 1958.

FV Webster. Traffic signals. *Road research technical paper*, 1966.

Hua Wei, Guanjie Zheng, Huaxiu Yao, and Zhenhui Li. Intellilight: A reinforcement learning approach for intelligent traffic light control. In *SIGKDD*, 2018.

Hua Wei, Chacha Chen, Guanjie Zheng, Kan Wu, Vikash Gayah, Kai Xu, and Zhenhui Li. Presslight: Learning max pressure control to coordinate traffic signals in arterial network. In *SIGKDD*, 2019a.

Hua Wei, Nan Xu, Huichu Zhang, Guanjie Zheng, Xinshi Zang, Chacha Chen, Weinan Zhang, Yanmin Zhu, Kai Xu, and Zhenhui Li. Colight: Learning network-level cooperation for traffic signal control. In *CIKM*, 2019b.

Yuanhao Xiong, Guanjie Zheng, Kai Xu, and Zhenhui Li. Learning traffic signal control from demonstrations. In *CIKM*, 2019.

Bingyu Xu, Yaowei Wang, Zhaozhi Wang, Huizhu Jia, and Zongqing Lu. Hierarchically and cooperatively learning traffic signal control. In *AAAI*, 2021.

Zhengxu Yu, Shuxian Liang, Long Wei, Zhongming Jin, Jianqiang Huang, Deng Cai, Xiaofei He, and Xian-Sheng Hua. Macar: Urban traffic light control via active multi-agent communication and action rectification. In *IJCAI*, 2020.

Xinshi Zang, Huaxiu Yao, Guanjie Zheng, Nan Xu, Kai Xu, and Zhenhui Li. Metalight: Value-based meta-reinforcement learning for traffic signal control. In *AAAI*, 2020.

Feng Zhang, Yani Liu, Ningxuan Feng, Cheng Yang, Jidong Zhai, Shuhao Zhang, Bingsheng He, Jiazao Lin, Xiao Zhang, and Xiaoyong Du. Periodic weather-aware lstm with event mechanism for parking behavior prediction. *IEEE Transactions on Knowledge and Data Engineering*, 2021.

Huichu Zhang, Siyuan Feng, Chang Liu, Yaoyao Ding, Yichen Zhu, Zihan Zhou, Weinan Zhang, Yong Yu, Haiming Jin, and Zhenhui Li. Cityflow: A multi-agent reinforcement learning environment for large scale city traffic scenario. In *WWW*, 2019.

Huichu Zhang, Markos Kafouros, and Yong Yu. Planlight: Learning to optimize traffic signal control with planning and iterative policy improvement. *IEEE Access*, 2020a.

Huichu Zhang, Chang Liu, Weinan Zhang, Guanjie Zheng, and Yong Yu. Generalight: Improving environment generalization of traffic signal control via meta reinforcement learning. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, 2020b.

Yu Zhang and Qiang Yang. A survey on multi-task learning. *IEEE Transactions on Knowledge and Data Engineering*, 2021.

Shengjia Zhao, Jiaming Song, and Stefano Ermon. Learning hierarchical features from deep generative models. In *ICML*. PMLR, 2017.

Guanjie Zheng, Yuanhao Xiong, Xinshi Zang, Jie Feng, Hua Wei, Huichu Zhang, Yong Li, Kai Xu, and Zhenhui Li. Learning phase competition for traffic signal control. In *CIKM*, 2019a.

Guanjie Zheng, Xinshi Zang, Nan Xu, Hua Wei, Zhengyao Yu, Vikash Gayah, Kai Xu, and Zhenhui Li. Diagnosing reinforcement learning for traffic signal control. *arXiv*, 2019b.

A APPENDIX

You may include other additional sections here.

Table 2: Implementation details of MTLIGHT

Items	Details
Number of policy steps	3600
Discount factor γ	0.95
Policy ϵ	$0.1 \rightarrow 0.01$
ϵ decay rate	0.995
Policy Learning rate	0.005
Policy minibatch	32
task-shared latent space dim	5
task-specific latent space dim	5
task-shared latent state coef	10
task-specific latent state coef	10
Policy network architecture	2 hidden layers, 20 nodes each, ReLU activations
Policy network optimizer	RMSprop with learning rate 0.001 and MSE loss
Multi-Task architecture	5 MLP embedding layers , 2 shared FC layers before GRU, GRU with hidden size 64, 1 shared FC layer after GRU, 4 task-specific FC layers, 4 output task layers ReLU activations
Multi-Task optimizer	Adam with learning rate 0.01 and MSE loss

B RELATED WORK

B.1 CONVENTIONAL AND ADAPTIVE TRAFFIC SIGNAL CONTROL

Most conventional traffic signal control methods are designed based on fixed-time signal control Webster (1958), actuated control Chiu (1992) or self-organizing traffic signal control Chiu & Chand (1993); Cools et al. (2013); Lowrie (1990); Svanes & Delaney (1981); Hunt et al. (1981). These approaches rely on expert knowledge and often perform unsatisfactorily in complicated real-world situations. To solve this problem, several optimization-based methods Roess et al. (2004); Varaiya (2013); Kouvelas et al. (2014) have been proposed to optimize average travel time, throughput, *etc.*, which decide the traffic signal plans according to the observed data instead of the human prior. However, these approaches typically rely on strict assumptions which might not hold in the real-world cases Webster (1966). Furthermore, the optimization problems are usually hard to tract and require significant computing power in complex scenarios.

B.2 RL-BASED TRAFFIC SIGNAL CONTROL

RL-based traffic signal control methods aim to learn the policy from interactions with the environment. Earlier studies use tabular Q-learning El-Tantawy et al. (2013); Abdoos et al. (2013); Dusparic & Cahill (2009); Abdoos et al. (2011) where the states in an environment are required to be discretized and low-dimensional. To address the unmanageable large or continuous state space, recent advances employ deep RL with more complex continuous state representations (like images or feature vectors) to map the high-dimensional states into actions.

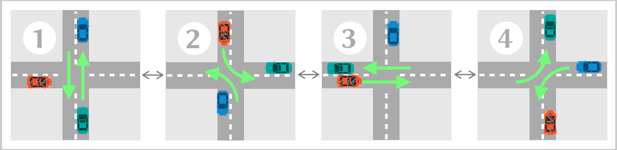


Figure 6: Illustration of phase.

Efforts have been made to design strategies that formulate the task as a single agent Wei et al. (2018); Mannion et al. (2016); Huang et al. (2021); Zang et al. (2020); Oroojlooy et al. (2020); Jiang et al. (2021); Rizzo et al. (2019) or some isolated intersections Zheng et al. (2019b;a); Xiong et al. (2019); Wei et al. (2019a); Chen et al. (2020); Oroojlooy et al. (2020); Zhang et al. (2020b;a), i.e., each agent makes decision for its own. The above methods are usually easy to scale, but they may have difficulty achieving globally optimal performance due to a lack of collaboration. To solve the problem, another way is to consider jointly modeling the action between learning agents with centralized optimization Van der Pol & Oliehoek (2016); Kuyer et al. (2008). However, as the number of agents increases, joint optimization usually leads to dimensional explosion, which has inhibited the widespread adoption of such methods to a large-scale traffic signal control. To overcome the difficulty, another type of methods are implemented in a decentralized manner, taking into account the collaboration between neighbors with appropriate reward and state design Arel et al. (2010); Nishi et al. (2018); Wei et al. (2019b); Xu et al. (2021). Methods such as El-Tantawy et al. (2013); Chu et al. (2019) add neighboring information into states, Nishi et al. (2018); Wei et al. (2019b); Yu et al. (2020); Guo et al. (2021) add neighbors' hidden features into states, and Xu et al. (2021) optimizes neighborhood travel time as an additional reward. However, simple concatenation of neighboring information is not reasonable enough because the influence of neighboring intersections is not balanced. Unlike the above methods that add neighbor information to the state, our method learns task-shared and task-specific latent states by constructing Multi-Task network.

B.3 Multi-Task Learning

Multi-Task Learning(MTL) Caruana (1997) is a learning paradigm aims to jointly learn multiple related tasks so that the knowledge contained in a task can be leveraged by other tasks. Past works Oh et al. (2017); Zhang & Yang (2021); Ruder (2017); Ndirango & Lee (2019) have found that, by sharing a representation among related tasks and jointly learning all the tasks, better generalization can be achieved over independently learning each task. Constructing auxiliary tasks to help the main task is a branch of Multi-Task Learning. Reinforcement learning is known to be sample inefficient, transferring knowledge from other auxiliary tasks is a powerful tool for improving the learning efficiency Jaderberg et al. (2016); Lin et al. (2019); Lyle et al. (2021); Tongloy et al. (2017); Bellemare et al. (2019). Lin et al. (2019) combines different auxiliary tasks which provide gradient directions to speed up the training of the main reinforcement learning task. In comparison, our work aims to transfer knowledge from the task-related auxiliary tasks as a prior to the main reinforcement learning task, to ultimately boost the performance. Specifically, we model the Multi-Task network as a latent structure where the task-shared latent state is generated from early layers and the task-specific latent state is generated from deeper layers. This incentivies the policy to learn the Bayers-optimal behaviours: the policy can take into account its uncertainty over the comprehensive information when choosing actions.

B.4 Preliminaries

In this section, we first introduce some basic concepts related to traffic signal control (TSC) that have been widely recognized in previous work Wei et al. (2019b); Zheng et al. (2019a); Zhang et al. (2020b); Wei et al. (2019a); Chen et al. (2020); Zang et al. (2020). Note that the concepts can be easily generalized to other intersections with different structures.

- **Incoming/Outgoing Lanes.** The incoming lanes refer to the lanes where the vehicles are about to enter the intersection. It usually contains three basic types: ”left-turn”, ”straight” and ”right-turn” from inner to outer. The outgoing lanes refer to the lanes where the vehicles are about to leave the intersection.

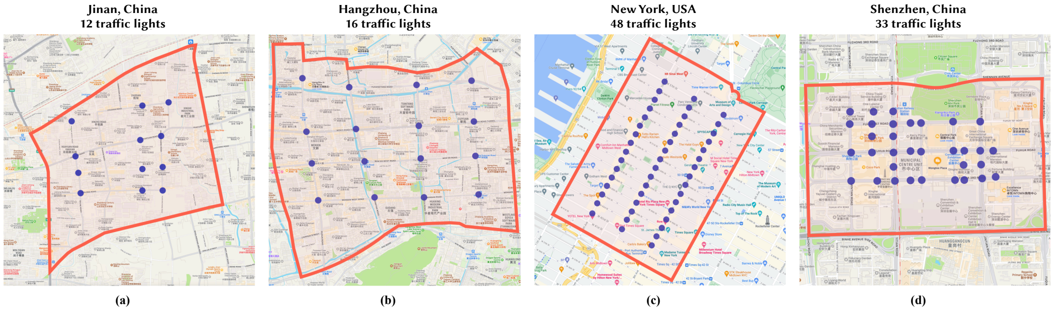


Figure 7: The illustration of the road networks. The figures from left to right represent the road network of Jinan(China), Hangzhou(China), New York(USA) and Shenzhen(China), containing 12 (4 × 3), 16 (4 × 4), 48 (16 × 3) and 33 (Non-grid) traffic signals respectively.

- **Roadnet.** A roadnet is a part of a dataset that represents an area of a city. A roadnet consists of signalized intersections, unsignalized intersections, and lanes connecting the intersections. Generally, the lane lengths, number of lanes and relative locations of intersections vary from one roadnet to another.
- **Phase.** Phase is a controller timing unit associated with the control of one or more movements, representing the permulation and combination of different traffic flows. The 4-phase setting is the most common configuration in reality, illustrated in Fig. 6, but the number of phases can vary due to different intersection topologies (3-way, 5-way intersections, etc.).
- **Queue Length.** Queue length is the number of vehicles waiting at an intersection due to a red light. Vehicles on the incoming lane with a speed of less than 0.1m/s are considered to be waiting.
- **Average Travel Time.** The travel time of a vehicle is the time discrepancy between entering and leaving a particular area. Average travel time of all vehicles in a road network is the most frequently used measure to evaluate the performance of traffic signal control Wei et al. (2019b;a); Zhang et al. (2020b); Chen et al. (2020); Zheng et al. (2019a).
- **Flow Distribution.** Flow distribution is the distribution of traffic entering the road network, which is generally expressed by the arrival rate of vehicles, i.e., the volume of traffic entering the road network per unit time.
- **Vehicles on Road.** Vehicles on road indicate the running vehicle, i.e., vehicles that have entered the road network and have not reached the end point. Vehicles on road can represent the real-time load on the road network.

C Algorithm

The algorithm is shown in Alg. 1.

D Datasets

D.1 Road Networks

The evaluation scenarios come from four real road network maps of different scales, including **Hangzhou** (China), **Jinan** (China), **New York** (USA) and **Shenzhen** (China), illustrated in Fig. 7. The road networks and data of Hangzhou, Jinan and New York are from the public datasets[1]. The road network map of Shenzhen is made by ourselves which is derived from OpenStreetMap[2]. The road networks of Jinan and Hangzhou contain 12 and 16 intersections in 4 × 3 and 4 × 4 grids, respectively. The road network of New York includes 48 intersections in 16 × 3 grid. The road network of Shenzhen contains 33 intersections, which is not grid compared to other three maps.

[1] https://traffic-signal-control.github.io/

[2] The road network map and data of Shenzhen will be released to facilitate the future research.

Algorithm 1: Training Process of MTLIGHT

```
Input: Roadnet file; traffic flow file; number of training episodes $E$; frequency of updating
       policy $t_p$; frequency of updating multi-task network $t_m$; total simulate time $T$
Output: Set of optimized parameters for the intersections; optimized parameter for the
        multi-task network
1  Initialize task-shared and task-specific latent state $\mathbf{o}_\mathrm{t}^\mathrm{shr}$, $\mathbf{o}_\mathrm{t}^\mathrm{spe}$
2  Initialize policy replay buffer $\mathcal{B}^\pi$
3  Initialize policy $\pi^\theta$ and multi-task network $\mathbf{M}^\phi$
4  Initialize reward of each agent $\{r_i \mid i \in 1, \ldots, n\}$
5  for episode ⟵ 1, 2, ..., $E$ do
6      for step t ⟵ 1, 2, ..., $T$ do
7          Collect original observations for all agents
8          Add task-shared $\mathbf{o}_\mathrm{t}^\mathrm{shr}$ and task-specific $\mathbf{o}_\mathrm{t}^\mathrm{spe}$ latent state to the observations
9          for agent i ⟵ 1, 2, ..., n do
10             Select action according to $\pi^\theta$
11         Employ joint action $\boldsymbol{a}$ to the environment
12         Get new observations and environmental reward
13         Collect trajectories to replay buffer $\mathcal{B}^\pi$
14         Get multi-task network input $\mathbf{f}_t^v, \mathbf{f}_t^s, \mathbf{f}_t^c, \mathbf{f}_t^{tr}, \mathbf{f}_t^q, \mathbf{f}_t^{vr}$ from the environment
15         Predict results using multi-task network $\mathbf{M}^\phi$
16         Get task-shared $\mathbf{o}_\mathrm{t}^\mathrm{shr}$ and task-specific $\mathbf{o}_\mathrm{t}^\mathrm{spe}$ latent state from $\mathbf{M}^\phi$
17         Calculate statistics from 0 up to $t$ as supervised signal
18         if t = $t_p$ then
19             Train policy $\pi^\theta$ by maximizing reward in Eq. 11
20             Clean up $\mathcal{B}^\pi$
21         if t = $t_m$ then
22             Calculate loss from the results of step 15 and step 17
23             Train multi-task network $\mathbf{M}^\phi$
24         if t = $T$ then
25             Collect the average total travel time of all vehicles as criteria
```

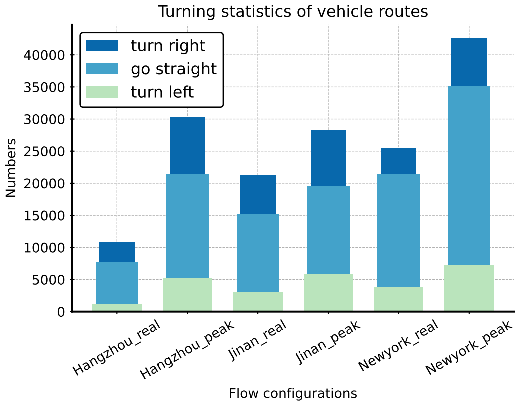


Figure 8: Turning statistics of vehicle routes.

D.2 FLOW CONFIGURATIONS

We run the experiments under two traffic flow configurations: real traffic flow and synthetic traffic flow. The real traffic flow is real-world hourly statistical data with slight variance in vehicle arrival rates, as shown in Tab. 3. Since the real-world strategies tend to break down during bottleneck period (peak hour), to better evaluate the performances of traffic light control methods in the flat-

Table 3: Arrival rate of real-world traffic dataset

Dataset	# Intersections	Arrival rate (vehicles/300s)			
		Mean	**Std**	**Max**	**Min**
$\mathcal{D}_{Hangzhou}$	16 (4 × 4)	248.58	42.25	333	212
$\mathcal{D}_{Jinan}$	12 (4×3)	524.58	102.91	672	256
$\mathcal{D}_{NewYork}$	48 (16×3)	235.33	5.84	244	224
$\mathcal{D}_{Shenzhen}$	33 (Non-grid)	147.92	79.35	255	22

Table 4: Data statistics of synthetic traffic dataset

Dataset	**Time**	**Arrival rate (vehicles/s)**	**Incoming vehicles**	**Accumulated vehicles**
$\mathcal{D}_{Hangzhou}$/ $\mathcal{D}_{Jinan}$/ $\mathcal{D}_{NewYork}$/ $\mathcal{D}_{Shenzhen}$	0-600	1.00	600	600
	600-1200	0.25	150	750
	1200-1800	4.00	2400	3150
	1800-2400	2.00	1200	4350
	2400-3000	0.2	120	4470
	3000-3600	0.5	150	4770

peak-flat scenario, we use synthetic datasets, which have a more dramatic variance in vehicle arrival rates, as shown in Tab. 4. A detailed description of traffic flow configurations is:

- ***Real.*** The traffic flows of **Hangzhou** (China), **Jinan** (China) and **New York** (USA) are from the public datasets, which are processed from multiple sources. The traffic flow of **Shenzhen** (China) is made by ourselves generated based on the traffic trajectories collected from 80 red-light cameras and 16 monitoring cameras in a hour. The data statistics are listed in Tab. 3.
- ***Synthetic.*** The *Synthetic* is a mixed traffic flow with a total flow of 4770 in one hour, to simulate a heavy peak. The arrival rate changes every 10 minutes, which is used to simulate the uneven traffic flow distribution in the real world, the details of the vehicle arrival rate and cumulative traffic flow are shown in Tab. 4.

E Evaluation Criteria

Following existing studies Wei et al. (2019b;a); Xiong et al. (2019); Chen et al. (2020); Zang et al. (2020), we use the **average travel time** to evaluate the performance of different methods for traffic signal control. The average travel time indicates the overall traffic situation in an area over a period of time. For a detailed definition of average travel time, see Section B.4. Since the number of vehicles and the origin-destination (OD) positions are fixed, better traffic signal control strategies result in less average travel time.

F Baselines

Our method is compared with the following two categories of methods: conventional transportation methods and RL methods[3]. Note that for a fair comparison all the RL methods are learned without any pre-trained parameters and the methods are evaluated under the same settings. The results are obtained by running the source codes[4]. All the baselines are run with three random seeds, and the mean is taken as the final result. The action interval is five seconds for each method, and the horizon is 3600 seconds for each episode. Specifically, the compared methods contain:

[3] Some existing RL based traffic signal control methods, such as AttendLight Oroojlooy et al. (2020) and SD-MaCAR Guo et al. (2021), evaluate their method under different experimental settings (e.g., road network or traffic flow), and the source codes are not available yet. Therefore, they are not compared in our experiments.

[4] https://github.com/traffic-signal-control/RL_signals

Table 5: Statistics of turning frequency at intersections in all routes.

Model	Hangzhou		Jinan		Newyork	
	real	syn_peak	real	syn_peak	real	syn_peak
turn left	1093 (14%)	5175 (24%)	3044 (20%)	5833 (30%)	3886 (18%)	7169 (20%)
go straight	6620 (86%)	16293 (76%)	12175 (80%)	13704 (70%)	17498 (82%)	27976 (80%)
turn right	3184	8752	5972	8747	4021	7421

F.1 CONVENTIONAL METHODS

- **MAXPRESSURE Varaiya (2013)** is a leading conventional method, which greedily chooses the phase with the maximum pressure. The pressure is defined as the difference of vehicle density between the incoming lane and the outgoing lane, and the vehicle density means the actual number of vehicles divided by the maximum permissible vehicle number.
- **FIXEDTIME Koonce & Rodegerdts (2008)** with random offset Roess et al. (2004) executes each phase in a phase loop with a pre-defined span of phase duration, which is widely used for steady traffic.
- **SOTL Cools et al. (2013)** specifies a pre-defined threshold for the number of waiting vehicles on approaching lanes. Once the waiting vehicles exceeds the threshold, it will switch to the next phase.

F.2 RL-BASED METHODS

- **INDIVIDUAL RL. Wei et al. (2018)** Independent control is performed for each agent in multi-agent environment, each intersection is controlled by one agent. The replay buffer and network parameters are not shared, and the model update is independent. There is no information transfer between agents, and no neighbor information is considered.
- **METALIGHT Zang et al. (2020)** is a value-based meta reinforcement learning method via parameter initialization, which is based on MAML Finn et al. (2017). METALIGHT is originally a single-agent approach for meta-learning on multiple separate tasks. Here we extend it to a multi-agent scenario without considering neighbor information.
- **PRESSLIGHT Wei et al. (2019a)** combines the traditional traffic method MAXPRESSURE Varaiya (2013) with RL technology together. PRESSLIGHT is a RL method that optimizes the pressure of each intersection.
- **COLIGHT Wei et al. (2019b)** uses graph convolution and attention mechanism to model the neighbor information, and then further uses this neighbor information to optimize the queue length.
- **GENERALIGHT Zhang et al. (2020b)** is a meta reinforcement learning method which uses generative adversarial network to generate diverse traffic flows and uses them to build training environments.