MetaLight: Value-based Meta-reinforcement Learning for Traffic Signal Control

Xinshi Zang[1], Huaxiu Yao[2], Guanjie Zheng[2], Nan Xu[1], Kai Xu[3], Zhenhui Li[2]
[1]Shanghai Jiao Tong University, [2]Pennsylvania State University, [3]Shanghai Tianrang Intelligent Technology Co., Ltd
[1]zang-xs@foxmail.com, [2]{huaxiuyao, gjz5038, jessieli}@ist.psu.edu, [1]xunannancy@sjtu.edu.cn, [3]kai.xu@tianrang-inc.com

Abstract

Using reinforcement learning for traffic signal control has attracted increasing interests recently. Various value-based reinforcement learning methods have been proposed to deal with this classical transportation problem and achieved better performances compared with traditional transportation methods. However, current reinforcement learning models rely on tremendous training data and computational resources, which may have bad consequences (e.g., traffic jams or accidents) in the real world. In traffic signal control, some algorithms have been proposed to empower quick learning from scratch, but little attention is paid to learning by transferring and reusing learned experience. In this paper, we propose a novel framework, named as MetaLight, to speed up the learning process in new scenarios by leveraging the knowledge learned from existing scenarios. MetaLight is a value-based meta-reinforcement learning workflow based on the representative gradient-based meta-learning algorithm (MAML), which includes periodically alternate individual-level adaptation and global-level adaptation. Moreover, MetaLight improves the-state-of-the-art reinforcement learning model FRAP in traffic signal control by optimizing its model structure and updating paradigm. The experiments on four real-world datasets show that our proposed MetaLight not only adapts more quickly and stably in new traffic scenarios, but also achieves better performance.

1 Introduction

Inefficient traffic signal plans waste people's time on roads. Current traffic signal control systems are not optimized according to the dynamic traffic data. For example, widely-adapted traffic control systems, such as SCATS (Lowrie 1992), rely on manually designed traffic signal plans. With the development of AI technology and the growth of available traffic data (e.g., surveillance camera data), recent studies apply deep reinforcement learning (DRL) on traffic signal control problems (Wei et al. 2018; Zheng et al. 2019a; Van der Pol and Oliehoek 2016). DRL methods can learn and adjust traffic signal policies based on the feedback from the environment and have shown better performance than traditional transportation methods.

The training mechanism of DRL follows a trial-and-error manner and thus the superior performance is conditioned on a large number of training episodes. The cost of computational resources and learning time is unacceptable in real-world traffic signal control. For example, if the traffic condition is complicated, traditional DRL models need long time to generate enough samples and to have models well-trained. Even worse, some successive bad trials may result in severe traffic congestion, which may break down the transportation system. Thus, the agent for traffic signal control should be able to learn quickly with a few samples.

Recently, meta-reinforcement learning has been widely studied to improve the efficiency of deep reinforcement learning by transferring previous learned knowledge and integrating this knowledge with the new information. There are mainly two lines of meta-reinforcement learning algorithms: (1) recurrent-based meta-reinforcement learning (Duan et al. 2016; Mishra et al. 2018). In this case, the parameters of the prediction model are controlled by a learnable recurrent meta-optimizer and its corresponding hidden state. (2) Gradient-based meta-reinforcement learning (Finn, Abbeel, and Levine 2017; Nagabandi, Finn, and Levine 2019; Nagabandi et al. 2019). These methods learn a well-generalized initialization that can be quickly adapted to a new scenario with a few gradient steps. However, simply applying either gradient-based or recurrent-based meta-reinforcement learning methods on traffic signal control faces two key challenges:

- **How to learn and adapt to the complicated and heterogeneous scenarios in traffic signal control?** Compared with previous meta-reinforcement learning applications that mainly focus on homogeneous tasks, the scenarios of traffic signal control are more complicated and heterogeneous. For example, the number of signal phases in different intersections varies from two to eight and one intersection may contain different numbers of lanes and roads. Since the DRL models in different scenarios are different, a sufficiently flexible meta-reinforcement learning model is required to handle various scenarios.
- **How to apply meta-learning on value-based reinforcement learning?** The action space for the traffic signal agent is discrete and small. For example, according

to (Wei et al. 2019c), the number of signal phases is usually no more than eight. With the small action space, value-based DRL is more suitable and it is more frequently used in current DRL-based traffic signal control (Wei et al. 2018), which trains the model in an off-policy fashion. However, current meta-reinforcement learning mainly focuses on policy-based DRL, where the on-policy data is used.

To address these challenges, we propose a novel meta-reinforcement learning framework for traffic signal control, **MetaLight**, which is built upon the gradient-based meta-reinforcement learning line. To the best of our knowledge, it is the first work to introduce meta-reinforcement learning paradigm into DRL-based traffic signal control. In MetaLight, we first improve a structure-agnostic DQN-based traffic signal control model called FRAP (Zheng et al. 2019a), which enables heterogeneous scenarios sharing the same parameters. Then, based on the meta-reinforcement learning paradigm, we learn a well-generalized initialization from various traffic signal control tasks. Given a new traffic scenario with a limited learning period, the learned initialization can be quickly adapted with a few generated samples. To address the second challenge, we further propose two types of adaptation mechanisms: individual-level adaptation and global-level adaptation. The former is a step-by-step optimization process on each task and the latter is a periodic synchronous updating process on a batch of sampled tasks. Each task inherits a globally-shared initialization of parameters, then performs individual-level adaptation and finally contributes to global-level adaptation.

We conduct extensive experiments to evaluate MetaLight on four real-world datasets. The results show that our proposed MetaLight enhances the learning efficiency and outperforms state-of-the-art baselines in traffic signal control. In summary, this paper has the following key contributions:

- To improve the efficiency of traffic signal control, we are the first to apply value-based meta-reinforcement learning for traffic signal control.
- We propose MetaLight, a novel value-based meta-reinforcement learning framework by combining individual-level adaptation and global-level adaptation.
- Empirically, we demonstrate the effectiveness and efficiency of our proposed model on four real-world datasets.

2 Related Work

Meta-reinforcement learning. Meta reinforcement learning aims to solve a new reinforcement learning task by leveraging the experience learned from a set of similar tasks. Currently, meta-reinforcement learning can be categorized into two different groups. The first group approaches (Duan et al. 2016; Wang et al. 2016; Mishra et al. 2018) use an external memory to store previous learned knowledge and further reuse these knowledge in a future task. For example, (Wang et al. 2016) trains a recurrent neural network by using the training data as input and then output the parameters of a leaner model. These approaches can achieve relatively good performances, but they may lack computational efficiency (Finn and Levine 2017).

In contrast, the second type of approaches (Li and Malik 2016; Finn, Abbeel, and Levine 2017; Nagabandi et al. 2019; Andrychowicz et al. 2016; Yao et al. 2019) aim to learn an optimal parameter initialization or optimizer. Representatively, model-agnostic meta-learning (MAML) (Finn, Abbeel, and Levine 2017) optimizes the initial parameters of the base learner in meta-training process, which significantly improves the efficiency of reinforcement learning on the new task. However, most gradient-based reinforcement learning algorithms are mainly focusing on policy-based reinforcement learning. How to combine MAML with value-based reinforcement learning is rarely studied.

Reinforcement learning for Traffic signal control. RL-based traffic signal control has attracted widely attention from both academia and industry in the last two decades. Traditional RL methods (Balaji, German, and Srinivasan 2010; Abdulhai, Pringle, and Karakoulas 2003) are limited to tabular Q-learning and a discrete state representation. However, with the development of RL methods, researchers have studied different RL methods in traffic signal control. In terms of algorithms, current studies can be categorized into value based methods (e.g., deep Q-Network (Van der Pol and Oliehoek 2016; Wei et al. 2019a; 2019b; Zheng et al. 2019b)) and policy-based methods (Aslani, Mesgari, and Wiering 2017; Xiong et al. 2019).

In addition to the different method category, researchers have also been exploring different design of the network and features. Early studies (Abdoos, Mozayani, and Bazzan 2011) use numerical features to describe traffic scenario, e.g., queue length of each lane. These features are fed into a multi-layer perceptron to predict the action (e.g., signal to set). Recently, researchers (Gao et al. 2017; Van der Pol and Oliehoek 2016) convert traffic situation features (e.g., positions of vehicles) into image, and apply convolutional neural networks (CNN) learn their representations. For instance, (Gao et al. 2017) successfully achieves nearly 50% improvements compared with transportation methods. Recently, (Wei et al. 2018) proposes a dual-branch network structure to effectively approximate value function. After that, (Zheng et al. 2019b) proposes a plain fully-connected neural net with concise state features and properly designed reward function, which outperforms all the state-of-the-art baseline methods.

However, one common problem of the aforementioned methods is the lack of a universal network design for different intersection scenarios, which means that we need to train different networks for different scenarios from scratch. (Zheng et al. 2019a) recently proposed a novel network design, called FRAP, based on the principle of phase competition, making it possible to apply universally to different intersections with the same set of network parameters.

In this paper, we make further modification based on FRAP to make it apply to more universal scenarios, including different lane and intersection settings. Additionally, we combine the improved FRAP ++ and the extended MAML paradigm in MetaLight to transfer the knowledge trained from different scenarios and enable quick adaptation to new scenarios.

3 Problem Statement

In this section, we first define several basic concepts and then formally define the meta-reinforcement learning problem for traffic signal control.

3.1 Preliminary

In this paper, we investigate traffic signal control in a single intersection with different scenarios. In most cases, the scenario of an intersection is determined by three concepts: traffic flow, entering approach or lane, and phase setting, which are explained as follows:

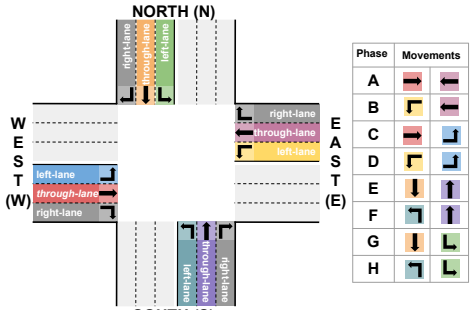


Figure 1: Intersection structure and traffic signal phase. (a) shows a standard intersection with four entering approaches (E/N/W/S), each of which has three types of lanes (right/through/left). (b) enumerates eight typical signal phases.

- **Traffic flow**: Both the pattern and volume of traffic flow are significantly different between intersections. In traditional DRL model for traffic light control, traffic flow is used as features, which does not change the state/action space (Wei et al. 2019c). Therefore, intersections only differing in traffic flows are regarded as homogeneous scenarios in this paper.
- **Entering approach/lane**: For each intersection, the entering approach is represented as the direction which vehicles enter in. In real world, most intersections are equipped with four entering approaches but some have three or even five. Figure 1 illustrates a standard 4-approach intersection. Each entering approach has three types of lanes, e.g., left-lane, through-lane and right-lane. According to (Wei et al. 2019c), many features in the state for RL methods are measured in unit of lanes, such as queue length per lane, the number of entering approaches and lanes determine the dimension of state space. Thus, intersections with different number of entering intersections and lanes are regarded as heterogeneous scenarios.
- **Phase Setting**: As illustrated in Figure 1, there are theoretically eight signal phase in total and each phase controls two traffic movements which do not conflict with each other. Each intersection has its own phase settings based on the traffic characteristics. Since the dimension of action space for RL agent is directly correlated with the number of phases (Wei et al. 2019c), we also define intersections with different phase settings as heterogeneous scenarios.

3.2 Problem: Meta-reinforcement Learning for Traffic Signal Control

Following the traditional task definition of meta-reinforcement learning (Finn, Abbeel, and Levine 2017), in traffic signal control, we are given a set of N_t intersections $\mathcal{I}_{\mathcal{S}} = \{\mathcal{I}_1, \ldots, \mathcal{I}_{N_t}\}$ sampled over task distribution $\mathcal{E}$. The control process in each intersection $\mathcal{I}_i$ is represented as a Markov decision process $\langle \mathcal{S}_i, \mathcal{A}_i, \mathcal{R}_i, \gamma_i, H_i \rangle$, which contains a finite set of states $\mathcal{S}_i$, a finite set of actions $\mathcal{A}_i$, a reward function $\mathcal{R}_i$, a discounted factor γ_i, and the episode length H_i. The reward $\mathcal{R}_i(s,a)$ in step t is defined as $\mathcal{R}_i(s,a) = \mathbb{E}[\mathcal{R}_{t+1} | \mathcal{S}_i(t) = s, \mathcal{A}_i(t) = a]$. For each intersection $\mathcal{I}_i$, given an episode length H_i, the goal is to learn an optimal control policy $\pi_i(a|s)$. In addition, for intersection $\mathcal{I}_i$, the value function is defined as the sum of reward r_t discounted by γ_i at each timestep t, which is formulated as

$$Q(s,a;f_\theta) = \mathbb{E}\left[r_i(t) + \gamma_i r_i(t+1) + \ldots | s_i(t) = s, a_i(t) = a\right]. \tag{1}$$

Then, we defined the base learner f with learnable parameter θ to map observations $\mathcal{S}_i$ to outputs $\mathcal{A}_i$. The effectiveness of function f with optimal parameters θ_i is defined as

$$\mathcal{L}(f_{\theta_i}) = \mathbb{E}_{s,a,r,s' \sim \mathcal{D}_i}\left[\left(r + \gamma \max_{a'} Q\left(s', a'; f_{\theta_i^-}\right) - Q\left(s, a; f_{\theta_i}\right)\right)^2\right], \tag{2}$$

where θ_i^- are the parameters of target network in FRAP that are fixed for every C iterations (Mnih et al. 2015).

In meta-reinforcement learning, we are supposed to learn a well-generalized meta-learner $\mathcal{M}(\cdot)$ to enhance the learning efficiency of future traffic signal control tasks. In general, the whole procedure of meta-learning can be split as two steps: meta-training and meta-testing. During meta-training, the parameters of base learner f (i.e., $\{\theta_1, \ldots, \theta_{N_t}\}$) and the well-generalized meta-learner $\mathcal{M}(\cdot)$ are updated alternatively. First, the parameters $\{\theta_1, \ldots, \theta_{N_t}\}$ are learned by using transitions $\mathcal{D}_i$ sampled from each intersection $\mathcal{I}_i$. The goal is to minimize the loss over all meta-training, which is defined as:

$$\{\theta_1, \ldots, \theta_{N_t}\} := \min_{\{\theta_1, \ldots, \theta_{N_t}\}} \sum_{i=1}^{N_t} \mathcal{L}(\mathcal{M}(f_{\theta_i}); \mathcal{D}_i). \tag{3}$$

Then, the meta-learner $\mathcal{M}$ is optimized by sampling another batch of transitions $\mathcal{D}_i^{'}$:

$$\mathcal{M} := \min_{\mathcal{M}} \sum_{i=1}^{N_t} \mathcal{L}(\mathcal{M}(f_{\theta_i}); \mathcal{D}_i^{'}). \tag{4}$$

After learning a well-generalized meta-learner, during meta-testing, for a new traffic intersection $\mathcal{I}_t$, the model f is adapted by using transitions $\mathcal{D}_t$ sampled from it.

Then, we introduce model-agnostic meta-learning (MAML), one of the representative gradient-based meta-reinforcement learning algorithms (Finn, Abbeel, and Levine 2017). In MAML, the meta-learner $\mathcal{M}$ is regarded as well-generalized initialization θ_0 of parameters in base learner f. With a few gradient descent steps, we can get the optimal parameters θ_i. Thus, the meta-learner $\mathcal{M}$ is regarded as (one gradient step as exemplary) $\mathcal{M}(f_{\theta_i}) = f_{\theta_0 - \alpha \nabla_\theta \mathcal{L}(f_\theta, \mathcal{D}_i)}$. In meta-training process, the whole loss of MAML is:

$$\mathcal{L}_{all} = \mathcal{L}(f_{\theta_0 - \alpha \nabla_\theta \mathcal{L}(f_\theta; \mathcal{D}_i)}; \mathcal{D}_i^{'}). \tag{5}$$

4 The MetaLight Framework

In this section, we first briefly introduce the structure-agnostic and parameter-sharing RL model called FRAP (Zheng et al. 2019a) and propose a improved model FRAP++. Then, we will elaborate the entire parameter learning procedure of our proposed MetaLight, including individual-level adaptation and global-level adaptation.

4.1 Structure-agnostic and Parameter-sharing RL Model

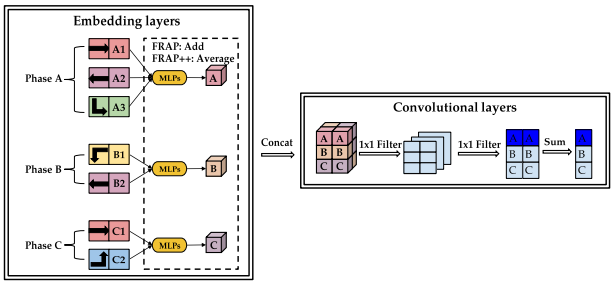


Figure 2: The Illustration of FRAP and FRAP++. FRAP uses the sum of lanes' representation to represent phase while FRAP++ uses the mean of them. Yellow multi-layer perceptrons (MLPs) are shared by each phase.

In traffic signal control, a flexible base model f is required to handle the scenario across heterogeneous intersections which are described in Sec. 3.1. Figure 2 illustrates structures of FRAP and FRAP++ in 3-phase intersections. The network consists of several embedding layers and convolutional layers. The former parameters are shared across lanes, which means the number and type of approaching lanes only affect the network structure rather than the parameters of embedding layers. Furthermore, FRAP uses fixed number of 1×1 filters in convolutional layers, they are also independent of the number and type of phase. In summary, the structure of FRAP depends on the number of lanes and phases in the intersection but the network parameters are sharing in different intersections.

To improve the flexibility of FRAP on different lanes combination, we propose a improved model FRAP++, which enhance FRAP from two folds: (1) The FRAP++ represents the phase demand by averaging each lane's demand instead of adding this demand in order to remove the influence of difference in the lane number under each phase and make FRAP widely applicable.. (2) FRAP updates parameters only after each whole episode, which violates DQN one-step updating mechanism. Instead, FRAP++ improves the updating frequency by undertaking a mini-batch updating after each step in one episode.

Similar with (Zheng et al. 2019a), the state of FRAP++ consists of the number of vehicles and signal phase on each approaching lane. The action for RL agent is defined as choosing the phase for the next time interval. The reward is defined as the average queue length on approaching lanes. Therefore, FRAP++ is a structure-agnostic model with shared parameters between different scenarios, which perfectly fits the property of base learner f defined in Sec. 3.

4.2 MetaLight Framework

Next, we introduce our MetaLight framework, which reuse previous learned knowledge to facilitate the learning process in target intersection. MetaLight follows the traditional gradient-based meta-reinforcement learning framework, MAML, which is described in Sec. 3. However, traditional design of MAML mainly focuses on policy-based DRL problems. Empirically, on value-based DRL models like FRAP++, MAML only slightly outperforms random initialization, which does not meet our expectation and cannot be deployed to large-scale real-world scenarios (see experiments in Section 5 for more details). Thus, we improve MAML by alternatively utilizing individual-level adaptation and global-level adaptation. Specifically, MetaLight takes advantage of fast learning in DQN by updating parameters at each time-step and extracting the common knowledge in MAML by gradient descent. The framework of MetaLight is illustrated in Figure 3 and we detail these two adaptation steps in the follows:

Individual-level Adaptation As described in (Mnih et al. 2015), DQN uses a neural network to represent the action-state function, $Q(s, a)$, in Equation (1). In traffic signal control, FRAP++ follows the standard design of DQN with experience replay and target value network. In each intersection $\mathcal{I}_i$, the agent's experiences $e_i(t) = (s_i(t), a_i(t), r_i(t), s_i(t+1))$ at each timestep t are stored in set $\mathcal{D}_i$.

As shown in Figure 3, in individual-level adaptation, the parameters θ_i of each task $\mathcal{T}_{i_s}$ are updated at each timestep by gradient descent, which is formulated as (one gradient step as exemplary):

$$\theta_i \leftarrow \theta_i - \alpha \nabla_\theta \mathcal{L}(f_\theta; \mathcal{D}_i), \tag{6}$$

where α represents the step size and the loss function $\mathcal{L}$ is defined in Eqn. (2). In value-based reinforcement learning, individual-level adaptation is taken at each timestep to speed up the learning process on source intersections.

Global-level Adaptation After the adaptation in individual-level, global-level adaptation aims to aggregate the adaptation of each intersection $\mathcal{I}_i$, and then update the initialization θ_0 of meta-learner using a newly sampled

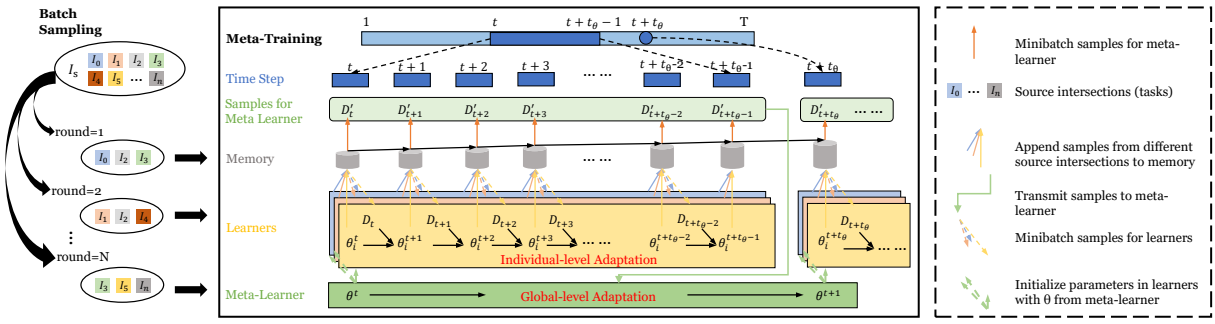


Figure 3: Meta-training framework of MetaLight. From left to right, a batch of tasks are first sampled. Then, in meta-training, the whole episode with a length of T is split by t_θ. During each interval t_θ, the base learner inherits the initialization from meta-learner and then conduct individual-level adaptation using samples drawn from memory at each time step. At the end of each interval t_θ, the meta-learner takes global-level adaptation with another batch of samples from the memory.

Algorithm 1: Meta-training process of MetaLight

Input: Set of source intersections $\mathcal{I}_\mathcal{S}$; stepsizes α, β frequency of updating meta parameters t_θ
Output: Optimized parameters initialization θ_0

Randomly initialize parameters θ_0
for *round* = $1, \ldots, N$ **do**
  Sample a batch of intersections from $\mathcal{E}$
  for $t = 1, t_\theta + 1, 2t_\theta + 1, \ldots, T$ **do**
    for $t^{'} = t, \ldots, \min(t + t_\theta, T)$ **do**
      for *each intersection* $\mathcal{I}_i$ **do**
        $\theta_i \leftarrow \theta_0$
        Generate transitions into $\mathcal{D}$ and sample transitions as $\mathcal{D}_i$
        Update $\theta_i \leftarrow \theta_i - \alpha \nabla_\theta \mathcal{L}(f_\theta; \mathcal{D}_i)$ by Eqn. (6)
    Sample new transitions from $\mathcal{D}$ as $\mathcal{D}_i^{'}$
    Update $\theta_0 \leftarrow \theta_0 - \beta \nabla_\theta \sum_{\mathcal{I}_i} \mathcal{L}(f_\theta; \mathcal{D}_i^{'})$ by Eqn. (7)

transitions $\mathcal{D}_i^{'}$. The initialization θ_0 is updated as follows:

$$\theta_0 \leftarrow \theta_0 - \beta \nabla_\theta \sum_{\mathcal{I}_i} \mathcal{L}(f_\theta; \mathcal{D}_i^{'}), \tag{7}$$

where β is defined as stepsize. The whole algorithm for meta-training process of MetaLight is described in Alg. 1.

Transfer Knowledge to New Intersections In the meta-training process of MetaLight, we learn a well-generalized initialization of parameters in f. Then, we apply the initialization θ_0 to a new target intersection $\mathcal{I}_t$. By using θ_0 as initialization, the update process in the intersection $\mathcal{I}_t$ is defined as:

$$\theta_t \leftarrow \theta_t - \alpha \nabla_\theta \mathcal{L}(f_\theta; \mathcal{D}_t). \tag{8}$$

Then we evaluate the performance by using the optimal parameters θ_t. The meta-testing process is outlined in Alg. 2.

Algorithm 2: Meta-testing process of MetaLight

Input: Set of target intersections $\mathcal{I}_\mathcal{T}$; stepsizes α learned initialization θ_0
Output: Optimized parameters θ_t for each intersection $\mathcal{I}_t$

for *each intersection* $\mathcal{I}_t$ *in* $\mathcal{I}_\mathcal{T}$ **do**
  $\theta_t \leftarrow \theta_0$
  for $t = 1, \ldots, T$ **do**
    Generate and sample transitions as $\mathcal{D}_t$
    Update $\theta_t \leftarrow \theta_t - \alpha \nabla_\theta \mathcal{L}(f_\theta; \mathcal{D}_t)$ by Eqn. (8)

5 Experiment

5.1 Experiment Settings

We conduct experiments[1] in a simulation platform called CityFlow (Zhang et al. 2019) [2], which provides the latest simulation environments for traffic signal control. The traffic data is first fed into the simulator and vehicles move to their destination according to the setting of the environment. The simulator executes the traffic signal actions from the control method and returns the state to the signal control method.

5.2 Datasets

We use four real-world datasets from two cities in China: Jinan (JN) and Hangzhou (HZ), and two cities in the United States: Atlanta (AT), and Los Angeles (LA). The raw traffic data from two Chinese cities contains the information about the vehicles coming through the intersections, which are captured by the nearby surveillance cameras. The other raw data from American cities is composed of the full vehicle trajectories which are collected by several video cameras along the streets[3]. Based on these raw data, we run the traffic flow for one hour and the entering lanes only consist of left-lane and through-lane.

[1]Codes are provided at https://traffic-signal-control.github.io/
[2]https://cityflow-project.github.io
[3]https://ops.fhwa.dot.gov/trafficanalysistools/ngsim.htm

Because of the limited kinds of phase setting in the raw data, we add some new phase settings in order to build enough heterogeneous scenarios. There are eleven kinds of phase settings in total, including four kinds of 4-phase, six kinds of 6-phase, and one 8-phase. They are divided into two groups named as PS1 and PS2 respectively. As described in Figure 4, PS1, colored red, contains six kinds of phase settings and PS2 colored blue consists of the other five phase settings.

Figure 4: Eleven phase settings in experiments are composed of different phases from A to H. Red represents PS1 and blue denotes PS2.

As summarized in Table 1, we construct 24 scenarios in Hangzhou as training set. The phase setting of each scenario is drawn from PS1. The testing set is classified into three types and introduced as follows: Task-1 is a set of homogeneous tasks in which testing sets are similar with training sets except traffic flow. Task-2 represents heterogeneous tasks which means testing datasets are different from training datasets in both traffic flow and phase setting. Task-3 consists of both homogeneous and heterogeneous tasks from different cities (Jinan, Atlanta, and Los Angeles).

Table 1: Summary of datasets

Datasets	Training Sets	Testing Sets		
		Task-1	Task-2	Task-3
Scenarios	26	6	5	16
Cities	HZ	HZ	HZ	JN/AT/LA
Phase Settings	PS1	PS1	PS2	PS1/PS2

5.3 Methods for Comparison

To evaluate the effectiveness and efficiency of our MetaLight, we compare it with several representative methods described as follows. All baselines use FRAP as the base model.

- **Random** : Random uses random initialization and train FRAP++ model from scratch.
- **Pretrained** : Pretrained means selecting one existing FRAP++ model's parameters as the initial parameter for a new intersection. The similarity of different intersections determines which model to be chosen. When in homogeneous setting, the model trained at the same phase setting is chosen. In heterogeneous setting, since there are no existing intersections with the same phase setting, the model trained at 8-phase setting will be used for initialization.
- **MAML** (Finn, Abbeel, and Levine 2017): In MAML, we combine the original framework of MAML reinforcement learning and FRAP. The original FRAP is greatly matched with MAML framework for policy-based reinforcement learning, because it also conducts model updating at the end of a whole episode.
- **SOTL** (Cools, Gershenson, and Hooghe 2013) Self-Organizing Traffic Light Control (SOTL) provides reference value for comparison, which is a classical transportation method. SOTL sets a pre-defined threshold for the number of waiting vehicles on approaching lanes and changes signal phases when the threshold is exceeded.

5.4 Model Details and Hyperparameter Settings

In MetaLight, the base model, FRAP++ shares the similar network structure with FRAP (Zheng et al. 2019a), except for the average operation in the embedding layers. The learning rates of learner and meta-learner are set as 0.001 for MetaLight and MAML in both meta-training and meta-testing. The episode length for all scenarios is 3600 seconds and the interval of each interaction between simulator and RL agent is 10 seconds. For MetaLight, the learner conducts model updating after each interaction using 30 samples and only one epoch for training. Meta-learner updates itself at intervals of ten times of learners' updating. For MAML , the learner first undertakes one centralized updating at the end of each episode with 1000 samples and 100 epochs for training. Then, the meta-learner updates itself using new episodes each time.

5.5 Evaluation Metrics

We choose **travel time** as the evaluation metric, which is also the most frequently used measure to judge performance in the transportation field. This metric is defined as the average travel time that vehicles spend on approaching lanes (in seconds).

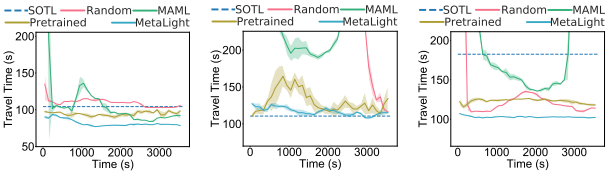
5.6 Task-1: Homogeneous Scenarios

In Task-1, we choose six homogeneous scenarios whose phase settings all come from PS1 and exist in the training set. The results of all methods are described in Table 2. Each phase setting stands for one scenario. Note that, the improvement is calculated by comparing with the best baseline. We can observe that either Pretrained or MAML is the best baseline but MetaLight outperforms them in most scenarios except for the 4b phase setting. The averaged improvement over these phase settings is 5.52%, which is not significant enough. The possible reason is that the effect of overfitting problem is not severe in homogeneous setting and simply utilizing existing models can work well. Even so, MetaLight is much better since it is able to apply only one initial model to all of these scenarios, while the Pretrained method need select suitable model each time.

Three meta-testing curves from these scenarios are illustrated here in Figure 5. In these cases, MetaLight outperforms other baselines and achieves not only faster learning speed but also the better converged value. Compared with MetaLight, MAML is not very good and sometimes becomes close to Random, which indicates the meta model trained in MAML can make few contributions to the learning in this problem.

Table 2: Performances of different methods on Task-1. Travel time is reported. The average improvement is 5.52%

Phase Setting	4a	4b	6a	6c	6e	8
Random	102.71	292.51	90.41	461.78	105.49	73.62
Pretrained	82.87	**191.83**	85.47	200.06	111.94	67.88
MAML	82.95	**191.53**	161.41	404.04	132.26	77.07
MetaLight	**74.67**	199.55	**78.92**	**195.92**	**98.58**	**66.93**
Improvement	9.89%	\	7.66%	2.07%	6.56%	1.41%



(a) Phase setting: 4a (b) Phase setting: 6a (c) Phase setting: 6e

Figure 5: Meta-testing results for Task-1. Travel time on each epoch is measured by testing the updated model in the whole episode. Three kinds of phase settings are selected from PS1. We exclude parts of curves which are out of range.

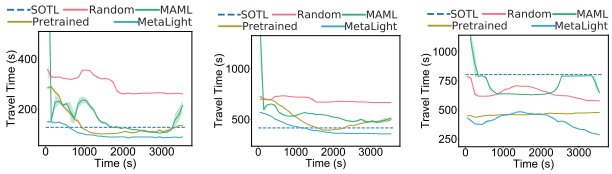
5.7 Task-2: Heterogeneous Scenarios

In Task-2, five heterogeneous scenarios in Hangzhou are selected. The results of these scenarios are shown in Table 3. We can see that MetaLight achieves significant improvements (22.57% in average). Because these phase settings in Task-2 are unseen in training datasets, the general knowledge shared by all scenarios could speed up learning in new scenarios and alleviate the impact of overfitting. Like Task-1, three learning curves of heterogeneous scenarios are shown in Figure 6.

We can see that MAML and Random adapt very slowly and cannot keep a stable learning trend, which means the initialization that MAML has learned is no better than random initialization. Because of the high variance rooted in policy-based RL, the original policy-based updating mechanism of FRAP and MAML bring too much unstable updating of the base model. Thus, it is hard to learn a universal initialization in traffic signal control. In contrast, MetaLight maintains a more stable and faster adaptation in new heterogeneous scenarios. Incorporating individual-level and global-level adaptation, MetaLight lets the base model to learn more stably and efficiently by finding an optimal universal initialization.

Table 3: Overall performances of Task-2. Each result is the average travel time of all scenarios. The averaged improvement over all phase settings is 22.57%.

Phase Setting	4c	4d	6b	6d	6f
Random	254.70	662.85	298.53	570.55	474.20
Pretrained	95.94	385.74	233.64	430.74	307.98
MAML	101.41	440.65	369.82	614.09	345.46
MetaLight	**81.07**	**352.83**	**172.91**	**273.58**	**226.82**
Improvement	15.50%	8.53%	25.99%	36.49%	26.35%



(a) Phase setting: 4c (b) Phase setting: 4d (c) Phase setting: 6d

Figure 6: Meta-testing curves for Task-2. Episode length is 3600s. Three random seeds are used for experiments. The means and variances of these results are also illustrated.

5.8 Task-3: Homogeneous and Heterogeneous Scenarios in Different Cities

Furthermore, we try to study the knowledge transfer between different cities. As described in Section 5.2, the source data may differ greatly, which increasing the difficulties to adapt control policy. We conduct homogeneous and heterogeneous experiments in Jinan, Atlanta, and Los Angeles. The results are presented in Table 4. Compared with Table 2, MetaLight significantly outperforms other baselines on homogeneous tasks. Figure 7 further illustrates detailed description of three learning curves for task-3, from which we can draw the same conclusion like Figure 5 and 6 that MetaLight outperforms all baselines and adapts much faster and more stable. Note that, in Figure 5, 6 and 7, the travel time may stay the same or rise for a while. This counter-intuitive phenomenons are mainly due to the randomness in the training process of RL model.

Table 4: Performances on Task-3. Each result is the average of three scenarios. Average improvements are 21.09% in homogeneous tasks and 9.59% in heterogeneous tasks.

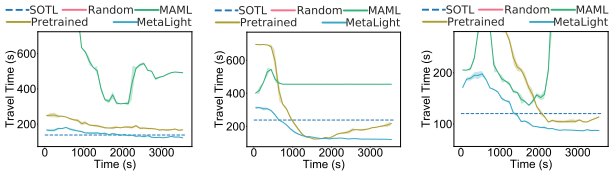
City	Homogeneous			Heterogeneous		
	JN	AT	LA	JN	AT	LA
Random	451.88	379.16	262.23	363.59	602.60	684.15
Pretrained	128.20	186.86	104.59	156.04	351.39	331.75
MAML	173.13	301.29	135.11	335.81	618.84	393.58
MetaLight	**95.01**	**161.37**	**77.23**	**137.02**	**310.39**	**308.71**
Improvement	25.89%	13.64%	26.16%	10.17%	11.67%	6.94%

6 Conclusion and Discussion

In this paper, we propose a novel framework MetaLight to improve the learning efficiency of deep reinforcement learning in traffic signal control by transferring previous

learned knowledge. We first improve a representative FRAP, a structure-agnostic traffic signal control model. Based on the previous gradient-based meta-learning framework, MetaLight then incorporates individual-level and global-level adaptation. The experiments on both homogeneous and heterogeneous scenarios demonstrate the effectiveness and efficiency of MetaLight for traffic signal control.

In the future, we plan to investigate this problem from the following two perspectives: (1) We plan to apply meta-reinforcement learning on traffic signal control across multi-intersections. In cooperation mechanism, the way to transfer the knowledge learned from existing scenarios need be carefully designed. (2) We plan to explain the black-box meta-reinforcement learning model by analyzing which knowledge is transferred.



(a) Jinan-6a (b) Atlanta-4d (c) Los Angeles-4c

Figure 7: Meta-testing results for task-3. The random baseline curves are excluded since their results are much worse.

Acknowledgements

The work was supported in part by NSF awards #1652525 and #1618448. The views and conclusions contained in this paper are those of the authors and should not be interpreted as representing any funding agencies.

References

Abdoos, M.; Mozayani, N.; and Bazzan, A. L. 2011. Traffic light control in non-stationary environments based on multi agent q-learning. In *2011 14th International IEEE conference on intelligent transportation systems (ITSC)*.

Abdulhai, B.; Pringle, R.; and Karakoulas, G. J. 2003. Reinforcement learning for true adaptive traffic signal control. *Journal of Transportation Engineering* 129(3):278–285.

Andrychowicz, M.; Denil, M.; Gomez, S.; Hoffman, M. W.; Pfau, D.; Schaul, T.; Shillingford, B.; and De Freitas, N. 2016. Learning to learn by gradient descent by gradient descent. *NIPS*.

Aslani, M.; Mesgari, M. S.; and Wiering, M. 2017. Adaptive traffic signal control with actor-critic methods in a real-world traffic network with different traffic disruption events. *Transportation Research Part C: Emerging Technologies*.

Balaji, P.; German, X.; and Srinivasan, D. 2010. Urban traffic signal control using reinforcement learning agents. *IET Intelligent Transport Systems* 4(3):177–188.

Cools, S.; Gershenson, C.; and Hooghe, B. D. 2013. Self-organizing traffic lights: A realistic simulation. *International Conferences on Self-Adaptive and Self-Organizing Systems*.

Duan, Y.; Schulman, J.; Chen, X.; Bartlett, P. L.; Sutskever, I.; and Abbeel, P. 2016. Rl2: Fast reinforcement learning via slow reinforcement learning. *arXiv preprint arXiv:1611.02779*.

Finn, C.; Abbeel, P.; and Levine, S. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning*.

Finn, C., and Levine, S. 2017. Meta-learning and universality: Deep representations and gradient descent can approximate any learning algorithm. *arXiv preprint arXiv:1710.11622*.

Gao, J.; Shen, Y.; Liu, J.; Ito, M.; and Shiratori, N. 2017. Adaptive traffic signal control: Deep reinforcement learning algorithm with experience replay and target network. *arXiv preprint arXiv:1705.02755*.

Li, K., and Malik, J. 2016. Learning to optimize. *arXiv preprint arXiv:1606.01885*.

Lowrie, P. 1992. Scats–a traffic responsive method of controlling urban traffic. roads and traffic authority. *NSW, Australia*.

Mishra, N.; Rohaninejad, M.; Chen, X.; and Abbeel, P. 2018. A simple neural attentive meta-learner. *ICLR*.

Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A. A.; Veness, J.; Bellemare, M. G.; Graves, A.; Riedmiller, M.; Fidjeland, A. K.; Ostrovski, G.; et al. 2015. Human-level control through deep reinforcement learning. *Nature*.

Nagabandi, A.; Clavera, I.; Liu, S.; Fearing, R. S.; Abbeel, P.; Levine, S.; and Finn, C. 2019. Learning to adapt in dynamic, real-world environments through meta-reinforcement learning. *ICLR*.

Nagabandi, A.; Finn, C.; and Levine, S. 2019. Deep online learning via meta-learning: Continual adaptation for model-based rl. *International Conference on Learning Representations*.

Van der Pol, E., and Oliehoek, F. A. 2016. Coordinated deep reinforcement learners for traffic light control. *Proceedings of Learning, Inference and Control of Multi-Agent Systems*.

Wang, J. X.; Kurth-Nelson, Z.; Tirumala, D.; Soyer, H.; Leibo, J. Z.; Munos, R.; Blundell, C.; Kumaran, D.; and Botvinick, M. 2016. Learning to reinforcement learn.

Wei, H.; Zheng, G.; Yao, H.; and Li, Z. 2018. Intellilight: A reinforcement learning approach for intelligent traffic light control. In *KDD*, 2496–2505.

Wei, H.; Chen, C.; Zheng, G.; Wu, K.; Gayah, V.; Xu, K.; and Li, Z. 2019a. Presslight: Learning max pressure control to coordinate traffic signals in arterial network.

Wei, H.; Xu, N.; Zhang, H.; Zheng, G.; Zang, X.; Chen, C.; Zhang, W.; Zhu, Y.; Xu, K.; and Li, Z. 2019b. Colight: Learning network-level cooperation for traffic signal control. CIKM.

Wei, H.; Zheng, G.; Gayah, V.; and Li, Z. 2019c. A survey on traffic signal control methods. *arXiv preprint arXiv:1904.08117*.

Xiong, Y.; Zheng, G.; Xu, K.; and Li, Z. 2019. Learning traffic signal control from demonstrations. CIKM, 2289–2292.

Yao, H.; Wei, Y.; Huang, J.; and Li, Z. 2019. Hierarchically structured meta-learning. In *International Conference on Machine Learning*, 7045–7054.

Zhang, H.; Feng, S.; Liu, C.; Ding, Y.; Zhu, Y.; Zhou, Z.; Zhang, W.; Yu, Y.; Jin, H.; and Li, Z. 2019. Cityflow: A multi-agent reinforcement learning environment for large scale city traffic scenario. *arXiv preprint arXiv:1905.05217*.

Zheng, G.; Xiong, Y.; Zang, X.; Feng, J.; Wei, H.; Zhang, H.; Li, Y.; Xu, K.; and Li, Z. 2019a. Learning phase competition for traffic signal control. CIKM, 1963–1972.

Zheng, G.; Zang, X.; Xu, N.; Wei, H.; Yu, Z.; Gayah, V.; Xu, K.; and Li, Z. 2019b. Diagnosing reinforcement learning for traffic signal control. *arXiv preprint arXiv:1905.04716*.