MetaSignal: Meta Reinforcement Learning for Traffic Signal Control via Fourier Basis Approximation

Shuning Huang, *Student Member, IEEE,* Kaoru Ota, *Member, IEEE,* Mianxiong Dong, *Member, IEEE,* and Huan Zhou *Member, IEEE*

***Abstract*—Traffic signal control plans significantly impact transportation system efficiency by regulating traffic conditions at intersections. Adaptive traffic plans that can adjust to real-time road conditions are more effective as a result. Reinforcement learning succeeds at adapting strategies based on feedback derived from the environment, and is thus proficient in dealing with complex traffic scenarios that change dynamically. However, current RL methods rely on significant computational periods to obtain precise functioning mechanisms within the scenarios, posing barriers to their adoption for new scenarios. In addition to directly optimizing the RL model itself to enable fast learning from scratch, another idea is to make the model transferable or reusable with the learned experience. Given the diversity of migration scenarios, the underlying control algorithm should guarantee convergence and endeavor to be parameter-insensitive. From the above concern, we proposed MetaSignal, an efficient meta-reinforcement learning method for traffic signal control. Specifically, our approach utilizes the Fourier basis as the value function approximation in reinforcement learning, distinguishing it from methods like neural network approximation. This linear approximation offers advantages such as convergence facilitation, error bound achievement, and reduced parameter dependence. The meta-learning framework adopts a model-agnostic approach, enabling effective adaptation of the base model to the target scenario with limited training cost. Empirically, the proposed method shows promising and stable performance for traffic signal control through comprehensive comparison experiments in both synthetic and real-world traffic networks.**

***Index Terms*—Meta-reinforcement learning, traffic signal control, function approximation.**

I. Introduction

Traffic signal control plans act as intersection controllers, indicating whether the movement is passable at the time through signals, and thereby influencing the efficiency of the traffic system. The dominant approach in current practice is to construct control plans based on historical observations of pavement conditions. Such plans have fixed patterns but have no flexibility to adjust to real-time conditions at intersections. Since the transportation environment features high dynamics, the ideal control methods are expected to be robust and adaptable, capable of tailoring the signals by actively interacting with the real-time environment [1]. Adaptive traffic signal control (ATSC) responds on demand, enhancing the system's throughput, and providing an optimized driving experience for vehicles on the transportation network by reducing average waiting times. Among the various approaches for its implementation, reinforcement learning (RL) proves to be more reliable [2].

RL-based ATSC methods learn the customized control policy suitable for road networks collaboratively by the multi-agents, with each intersection controlled by an autonomous agent. Notably, these methods do not provide agents with the mechanism underlying the system operation or the policies' role in influencing the system's states. Instead, agents are required to learn from the valid samples collected to acquire such knowledge. The process raises two concerns: First, there is a need for sufficient datasets for complex traffic conditions, as the number of training sets increase, the dimension of the state space grows exponentially. Second, owing to the trial-and-error nature of the training mechanism, a substantial amount of exploration time is indispensable to ensure the model's comprehensive training [3]. Nevertheless, these methods do not provide a universal structure design that can be adapted to all different road networks. When learning a new scenario, retraining the corresponding models from scratch is often necessary, the training cost is considerable.

To enhance the generalization of RL-based methods and improve efficiency, a helpful approach is to utilize the acquired knowledge to facilitate learning in target scenarios. This is where meta-learning emerges as a valuable strategy, aiming to get a well-generalized meta-learner capable of quickly adapting the specific base learner to the new scenarios [4]. Through meta-reinforcement learning, ATSC methods extract shared knowledge from a limited range of training scenarios and utilize this knowledge to initialize new learning tasks. Consequently, the exploration duration of the optimal solution space is significantly reduced, leading to quicker convergence to the optimum solution within fewer trials. Several studies have validated the feasibility of this scheme, but all of them are based on neural networks value function approximation [5]–[7].

To derive common knowledge applicable to diverse traffic environments, the underlying ATSC algorithm employed requires convergence guaranteed and parameter insensitivity, ensuring acceptable performance across encountered scenarios. Theoretical analysis [8] indicates that RL via linear function approximations exhibits manageable error bounds and stable convergence, whereas nonlinear cases may suffer from diver-

Shuning Huang, Kaoru Ota and Mianxiong Dong are with the Department of Sciences and Informatics, Muroran Institute of Technology, Muroran, Hokkaido, Japan. E-mail: {19096510, ota, mxdong}@mmm.muroran-it.ac.jp.

Huan Zhou is with college of Computer and Information Technology, China Three Gorges University, Yichang, China. E-mail: zhouhuan117@163.com.

gence in various situations. Moreover, linear learning methods require fewer parameters, resulting in higher generalizability across different tasks. Additionally, linear function approximation demonstrates high efficiency in terms of training data utilization and computational properties. However, it is essential that the value function to be linearly approximated must have guaranteed sufficient precision. Among various basis functions, the Fourier basis linear function approximation [9] stands out as it does not require transition samples, and the set of basis functions is constructed at a low cost. Furthermore, its structure incorporates prior knowledge about variable interactions, thus facilitating the induction of the value function's structure.

In this paper, we propose MetaSignal, a novel meta-reinforcement learning model for traffic signal control. To the best of the authors' knowledge, this is the first work that addresses meta-reinforcement ATSC learning via Fourier basis linear function approximation. MetaSignal follows MAML [4], [10], enables tailoring to new traffic scenarios by efficiently adapting the generic initialization model with restricted learning resources. As the acquired base model requires further adaptation to diverse transportation scenarios, Fourier basis function is employed to linearly approximate the RL value function, as it provides convergence guarantees with verifiable error bounds, in contrast to other non-linear methods [9], [11]. Besides, considering the classical Max Pressure control [12] has proved its sophistication in the RL domain [13], [14], we employ its variant *traffic intensity* for evaluating the state and reward of the RL agent [15]. Empirically, the diverse experiments demonstrate that the proposed method outperforms the representative baselines, while its base algorithm shows efficient performance. The technical contributions of this work are listed as follows:

- For the general information extracted needs to fit diverse transportation environments, we are the first solve the value function via Fourier basis function approximation for meta-reinforcement ATSC learning.
- The proposed base linear function approximation ATSC model shows the advantages of stable performance and rapid convergence over other methods.
- The conducted comparison experiments on synthetic and real-world traffic datasets validate the proposed control method's strong learning performance with sustained high efficiency.

The rest of this paper is arranged as follows. Section II introduces the recent development of the relevant technologies. The specification of the task environment and detailed definitions for RL is then given in Section III. Section IV presents the framework of MetaSignal, followed by Section V, which demonstrates its performance through comparative experiments on synthetic and real datasets. Lastly, Section VI offers concluding remarks.

II. Related Work

Most traditional traffic signal control methods are established control schemes based on human priori, are still widely used for traffic signal management in the current real world. These methods rely heavily on expert knowledge, unable to sense and react to real-time traffic situations, and typically perform unsatisfactorily in the face of complicated real-world scenarios. Subsequent conventional improvements have set it up as the optimization problem in certain traffic environments. For instance, [16] proposed their traffic signal control method based on stochastic predictive control, that incorporates uncertainty in exogenous traffic flow and downstream traffic flow turn rates in the modeling. [17] specifically considers the traffic demand for lane changing, incorporating additional features of lanes configuration beyond the typical traffic phase and vehicle trajectories, bringing it constructive to autonomous driving environments as well. Max Pressure (MP) [12] proposes a constructive traffic indicator named *pressure*, defined as the difference between the number of vehicles entering and exiting intersections, the value of which is positively correlated with intersection congestion, and the idea of which is still practically instructive. However, the strict assumptions and simplified dealing with traffic conditions these methods rely on pose obstacles to their real-world application.

In contrast, RL-based algorithms derive experiences directly from interactions with the environment instead of manually settings, and thus have higher potential. PressLight [13] implemented MP control [12] through reinforcement learning, followed MPLight [14] makes fine-tuned adjustments to the model's structure, enabling its application on a broader scope of traffic environments. Furthermore, IPDALight [15] incorporates the vehicle speed, position, and interaction between traffic intersections into the modeling based on Presslight, upgrades the concept of *pressure* to *intensity*, and distinguishes it from the conventional RL class of TSC algorithms by making the phase duration flexible.

Besides, plenty more studies present their ATSC logic from flavorful perspectives. FPAR [18] gives an expanded form of phase definition and assigns signals to different intersections by quantifying the prioritized demand between possible phases. [19] dynamic concurrence of computational resources by asynchronous advantage actor-critic learning, results in optimized control performance. Instead, [20] analyzes the priority of traffic flows through the main usage of directed acyclic graphs. In addition to optimizing the commute duration experience of vehicles on the road network, some research has focused on the wider needs of vehicles. PrivacySignal [21] considers the need for commuters to have their privacy adequately protected, adding secure interactive protocols along with the control method. SafeLight [22] designed its traffic signal control plans intending to ensure zero crashes at intersections by making road safety standards mandatory.

Since RL-based algorithms are trained based on data collected from the environment, they tend to be applicable only on particular scenarios. To extend the transferability of such algorithms, some attention has been paid to meta-learning, which transfers the knowledge gained from similar tasks to help the learning of a new task, bringing improvements in both learning efficiency and effectiveness. Model-agnostic meta-learning (MAML) [4] learns a set of optimized parameters with good representation and applies them to initialize the subsequent tasks, so that the tasks can be applied at a trivial learning

cost. It remain stands for the representative meta-learning algorithm. In practice, scholars have found that the idea of MAML is too idealized, just one set of global parameters is not enough to cope with complex and diverse environments: MUMOMAML [10] suggests learning the global initiation parameters for distinct environments separately. HSML [23] envisions summarizing different environments into finite groups by Cluster algorithms, and finds the corresponding global parameters for each cluster of tasks. Borrowing the above sophistication ideas, some works achieve meta-reinforcement learning of ATSC. MetaLight [5] adopts FRAP [18] as the base learner and follows the MAML framework [4] to implement meta-learning. Its undeniable drawback is that the method merely enables parameter mapping at only single intersections. Instead, GeneraLight [6] actualizes meta-learning as guided by HSML [23]. Accordingly, it adjusts its base learner centered on the clustering algorithm [24], and confirms the performance on the generative dataset where cluster categories can be clearly discerned. [7] employs the attention mechanism to aggregate the compelling features, proposing its DQN-based meta-learning approach.

The linear approximation of the RL value function has been researched through iterative. [9] shows the feasibility of using the Fourier basis as the linear continuous function approximation. TD(λ) [25] adapts the approximation to the need for true online time-difference (TD) learning to be updated at every step, instead of the traditional TD which only updated at the end of each episode, and proposes a new form of eligibility trace. [26] supports [25] with more insightful and substantial theoretical argumentation. TOS(λ)-FB [11] applies the Fourier basis as basis approximation functions to ATSC, and verifies the method's control effects on a real dataset in Cottbus, Germany based on the MATSim simulator. CycleRL [27] instead uses the Kalman filter as the linear approximation function, but can not control complicated traffic systems such as with unmodeled traffic features, thus only the single intersection dataset was taken as the target for the simulation.

III. Model Preliminaries

This section begins with introducing several fundamental concepts within the traffic environment, where traffic signal control is situated. Subsequently, we establish a standardized definition of the problem within the framework of meta-reinforcement learning.

A. Traffic Environment Definitions

This subsection describes the core transportation environment elements, where *intensity* is defined according to [15], and gives Fig. 1 to show the specific concepts visually.

· **Traffic movement:** A traffic movement involves the passage of vehicles from an entering lane to an exiting lane across an intersection. In a standard four-way entering configuration, as depicted in Fig. 1(a), there are 12 distinct traffic movements. Among these, traffic signals regulate all movements except right turns, as right-turning vehicles usually have the freedom to proceed without signal control, yet they must yield at a red light. It is pertinent to mention that in real-world intersections, a single traffic movement may encompass multiple lanes. However, our model simplifies this complexity by focusing on traffic signals controlling movements rather than specific lanes.

· **Signal Phase:** Uncoordinated traffic movement combinations, such as "east to west" and "south to north," can create chaotic road conditions at intersections, leading to reduced traffic efficiency and an increased likelihood of traffic accidents. A *phase* serves as a timing unit for the controller, defining a specific set of permissible pairs of traffic movements that represent various combinations of allowed traffic flows. Fig. 1(b) enumerates four detailed phase settings, which are commonly used in realistic traffic intersection scenarios. As an example, in Fig. 1(a), all three intersections activate phase #2, allowing vehicles in the straight east-west lanes to pass. Additionally, specific intersections adjust their phase settings based on their unique traffic intersection topologies (e.g., 3-way, 5-way intersections).

· **Intensity of vehicles:** Traffic intensity similar to pressure, also indicates the ability of a vehicle to indicate its contribution to the congestion level of the traffic flow, while capturing more dynamics than pressure. Vehicles closer to the intersection will bring more intensity to the intersection. Besides, their speed through the intersection negatively correlates with the intensity they bring. Thus, the intensity of a vehicle v is determined by the distance from its current location to the target intersection l_v, the current speed s_v, the length of the lane L in which it is located and the maximum speed limit $s_{\max}$:

$$T_v = \ln\left(1 + \frac{L - l_v}{L} \times \frac{s_{\max} - s_v}{s_v + 1}\right). \tag{1}$$

Given the variability and quantity of the above parameters, we directly label each vehicle in the Fig. 1(a) with its current exemplary intensity value.

· **Intensity of intersections:** The intensity $T_{\mathcal{I}_i}$ of a given intersection I_i reflects the intensity between vehicles entering and leaving the road network. It exhibits a negative correlation with the balance of vehicle distribution density at the intersection, with higher values indicating a greater imbalance in vehicle distribution. Take the situation shown in Fig. 1(a) as an example, the intensity values at each of the three intersections are $T_{I_1} = 3.6 = (0.8 + 1.3 + 1.3 + 1.6 + 0.8 - 1.2 - 1.0)$, $T_{I_2} = 3.5 = (1.2 + 1.0 + 1.7 + 0.8 + 1.2 - 1.6 - 0.8)$, and $T_{I_3} = 3.4 = (1.9 + 1.2 + 1.5 - 1.2)$.

· **Impacts of Neighboring Intersections:** Regard the intersections directly adjacent to the intersection in the corresponding four cardinal directions as their immediate neighbors. The impacts for neighboring intersection I_j to target intersection I_i are calculated as below, Notable, if there are no immediate neighbors in the intersection's given direction, its corresponding pressure is marked as 0:

$$P_{I_i \leftarrow I_j} = \omega \sum_{lane_i \in lane_{in}} \left(T_{lane_i} \times \min\left(\frac{\eta \times t}{N_{lane_i}}, 1\right)\right). \tag{2}$$

where ω, η are constant scalar, t marks the remaining time of the current phase. $lane_{in}$ indicates all lanes in the neighboring intersection I_j that can enter the target intersection

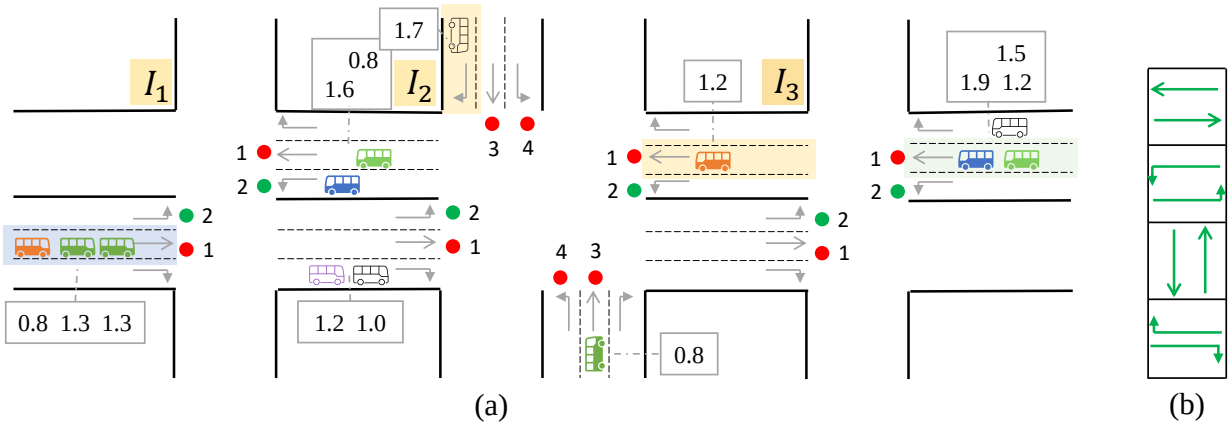


Fig. 1: Illustration of traffic environment definitions. (a) presents a schematic illustration of a 1x3 road network, where intersection I_2 is a standard intersection with 4-way entering approaches, each of which has three types of lanes (right/through/left). Some moving vehicles are symbolized with the current intensity, where all the dotted vehicles are located in the right-turn lanes, and their travel status are not controlled by signals. (b) enumerates four typical signal phases.

I_i, $N_l ane_i$ denotes the vehicles number on the incoming lane $lane_i$. For the example in Fig. 1(a), take $\omega = 0.5$, $\eta = 2$, $t = 1$ (unit: second), the blue and green boxes accordingly highlight vehicles entering I_2 from I_1 and I_3, while the yellow box marks vehicles entering I_1 from I_2. thus $P_{I_2 \leftarrow I_1} \approx 1.13 = 0.5 \times (3.4 \times \min(\frac{2\times1}{3}, 1))$, $P_{I_2 \leftarrow I_3} = 1.55 = 0.5 \times (3.1 \times \min(\frac{2\times1}{2}, 1))$, and $P_{I_1 \leftarrow I_2} = 1.45 = 0.5 \times (1.7 \times \min(\frac{2\times1}{1}, 1) + 1.2 \times \min(\frac{2\times1}{1}, 1))$. Since there is no vehicle trying to enter I_3 from I_2 , then $P_{I_3 \leftarrow I_2} = 0$.

B. Reinforcement Learning Environment

Suppose there are a total set of intersections, each of them is controlled by an RL agent. At time step t, based on the certain traffic situation, agents observe part of the environment from the finite state set $\mathcal{S}$, then chooses the optimal actions from the actions set $\mathcal{A}$ according to the policy $\pi(\mathcal{A}^t = a|\mathcal{S}^t = \boldsymbol{s})$. The reward function $\mathcal{R}$ is accordingly specialized as $\mathcal{R}(\boldsymbol{s}, a) = \mathbb{E}[\mathcal{R}|\mathcal{S}^t = \boldsymbol{s}, \mathcal{A}^t = a]$ in step t. Import the discounted factor γ, this control problem can be represented as a Markov Decision Process $\langle \mathcal{S}, \mathcal{A}, \pi, \mathcal{R}, \gamma \rangle$. One way to acquire the policy π can by estimate the optimal action-value function $Q^*(\boldsymbol{s}^t, a^t) = \max_\pi Q_\pi(\boldsymbol{s}^t, \pi(\boldsymbol{s}^t))$, where $\forall(\boldsymbol{s}^t, a^t) \in \mathcal{S} \times \mathcal{A}$. The concrete state design, action design, and reward design under current multi-agent learning are clarified below.

· **Observation:** The precise assessment of congestion levels in the approaching lanes holds paramount significance in making informed decisions about selecting the subsequent active signal phase in traffic signal control. Thus, each agent constitutes the observation state $\boldsymbol{s}_{\mathcal{I}_i}$ with the following components: (1) The current signal phase, available directly from the simulator. (2) The intensity of each phase and the traffic pressure from their immediate neighbors, are evaluated by the real-time state values obtained from the environment. The complete system state is formed by the collective observations of all agents.

· **Action:** At each timestep t, each agent executes a phase as its action a_t, which stands for the traffic signal setting should be used in period $t+1$ for intersection $\mathcal{I}_i$. Especially for this work, the entire phase set has four candidates, as indicated in Fig. 1(b). Notably, the phase selection is not guided by the cyclic or even-time principle, but the RL method chooses the best phase to set.

· **Reward:** Our method takes the intensity of intersection $T_{\mathcal{I}_i}$ as the reward of the according agent, which quantifies the degree of uneven vehicle distribution at the intersection based on the intensity difference between incoming and outgoing vehicles. For agent $\mathcal{I}_i$, its reward $r_{\mathcal{I}_i}$ is given by $r_{\mathcal{I}_i} = -T_{\mathcal{I}_i}$. Therefore, the rewards for the three intersections in Fig. 1(a) are $r_{\mathcal{I}_1} = -3.6$, $r_{\mathcal{I}_2} = -3.5$ and $r_{\mathcal{I}_3} = -3.4$. Minimizing the reward leads to more efficient utilization of green lights, ultimately enhancing the vehicle's travel experiences.

IV. Methodology

In general, MetaSignal learns a well-generalized meta-learner that enable speedy adaptation of the specific base learner to the target scenarios, thus enhancing the learning efficiency of approaching traffic signal control tasks. Next, we present in turn the base model, which explicitly implements the learning of traffic signal control by linear function approximation, and the meta-learner, which facilitates the knowledge gained by the extended MAML paradigm to the target intersections.

A. Base Linear Function Approximation Q-learning Model

To control the traffic signal, the base control method $f(\boldsymbol{\theta})$ receives state as input, predicts the Q-value for each action, optimizing policy by minimize the reward. Since the knowledge learned needs to be further transferred to other scenarios as communal knowledge, the underlying base ATSC algorithm employed requires convergence guaranteed and is non-parameter sensitive. In this paper, we use linear function to estimate the Q-value. Among the many basis functions, we adopt Fourier series for its predominance [9], [11].

Algorithm 1 Model-agnostic Meta-learning via Fourier Basis

Input: Set of source road networks $\mathcal{Y}_{\mathcal{S}}$; target road network $\mathcal{Y}_g$; stepsizes α, β; meta-parameter updating frequency coefficient $\tilde{t}$

Output: Optimized parameters $\boldsymbol{\theta}_g$ corresponding to $\mathcal{Y}_g$

* `Meta-Training` ;

Randomly initialize parameters $\boldsymbol{\theta}_0$;
for *training round=1,2,*$\cdots$ **do**
  for *t=1,* $\tilde{t}$*+1,* $2\tilde{t}$*+1,* $\cdots$*,T* **do**
    for t'*=1,* $\cdots$*, min(t+*$\tilde{t}$*, T)* **do**
      for *each* $\mathcal{Y}_i \in \mathcal{Y}_{\mathcal{S}}$ **do**
        $\boldsymbol{\theta}_i \leftarrow \boldsymbol{\theta}_0$;
        Compute coefficient vectors $\boldsymbol{c}_i$;
        Generate transitions into $\tilde{\mathcal{D}}$ from $\mathcal{Y}_i$;
        Sample a transitions set from $\tilde{\mathcal{D}}$ as $\mathcal{D}_{\mathcal{Y}_i}$;
        Compute features $\boldsymbol{\phi}_{\mathcal{Y}_i}(\boldsymbol{s}, a)$;
        Update $\boldsymbol{\theta}_{\mathcal{Y}_i} \leftarrow \boldsymbol{\theta}_{\mathcal{Y}_i} - \alpha \nabla_{\boldsymbol{\theta}} \mathcal{L}_{\mathcal{D}_{\mathcal{Y}_i}}(f_{\boldsymbol{\theta}})$;
      end for
    end for
    Sample a new set of transitions from $\tilde{\mathcal{D}}$ as $\mathcal{D}'_{\mathcal{Y}_i}$;
    Compute features $\boldsymbol{\phi}'_{\mathcal{Y}_i}(\boldsymbol{s}, a)$;
    Update $\boldsymbol{\theta}_0$ by $\boldsymbol{\theta}_0 \leftarrow \boldsymbol{\theta}_0 - \beta \sum_{\mathcal{Y}_i} \nabla_{\boldsymbol{\theta}} \mathcal{L}_{\mathcal{D}'_{\mathcal{Y}_i}}(f_{\boldsymbol{\theta}})$;
  end for
end for

* `Meta-Testing` ;

for $\mathcal{Y}_g$ **do**
  $\boldsymbol{\theta}_{\mathcal{Y}_g} \leftarrow \theta_0$;
  for *t=1, 2,* $\cdots$*,T* **do**
    Compute coefficient vectors $\boldsymbol{c}_g$;
    Generate and sample transitions as $\mathcal{D}_{\mathcal{Y}_g}$ from $\mathcal{Y}_g$;
    Compute features $\boldsymbol{\phi}_{\mathcal{Y}_g}(\boldsymbol{s}, a)$;
    Update $\boldsymbol{\theta}_{\mathcal{Y}_g}$ by $\boldsymbol{\theta}_{\mathcal{Y}_g} \leftarrow \boldsymbol{\theta}_{\mathcal{Y}_g} - \alpha \nabla_{\boldsymbol{\theta}} \mathcal{L}_{\mathcal{D}_{\mathcal{Y}_g}}(f_{\boldsymbol{\theta}})$;
  end for
end for

In the Fourier basis linear function approximation case, the action-value function $Q(\boldsymbol{s}, a)$ is approximated as a learned weights parameter vector $\boldsymbol{\theta} \in \mathbb{R}^m$ and a weighted sum of a set of m basis functions $\{\boldsymbol{\phi}_i(\boldsymbol{s}, a)\}_{i=1}^m$:

$$Q_\theta(\boldsymbol{s}, a) \approx \theta \cdot \boldsymbol{\phi}(\mathbf{s}, a). \tag{3}$$

$\boldsymbol{\phi}(\mathbf{s}, a) = \sum_{i=a}^m \boldsymbol{\phi}_i(\mathbf{s}, a)$ stands for the vector containing all m features for the state-action pair $(\mathbf{s}, a)$. The elements in $\boldsymbol{\phi}_{\mathcal{Y}_i}(\boldsymbol{s}, a)$ that correspond to the current action at take on the values of the Fourier basis, besides, the elements corresponding to other actions are all defined to be valued in zero:

$$\boldsymbol{\phi}_{\mathcal{Y}_i}(\boldsymbol{s}, a) = \begin{cases} cos(\pi \boldsymbol{c}_j \cdot \boldsymbol{s}) & \text{if } a = a^t \\ 0 & \text{otherwise} \end{cases}, \tag{4}$$

where $\mathbf{c}_j = [c_1, \cdots, c_{|s|}]$ denotes the coefficient vectors, each coefficient $c_{jk} \in \{0, \cdots, n\}$, $1 \leq j \leq m$, $1 \leq k \leq |s|$ determines the j-th basis function's frequency along the k-th state dimension. Practically, m is determined by how closely the method captures the interactions between the different state features. Here, we consider the interaction between at most two features. The order n of Fourier approximation stands for how large the frequency coefficients are basis functions considered, this work defines it as the dimension of state features.

Following the linear function approximation update rule, the weights $\boldsymbol{\theta}$ undergo updates via stochastic gradient descent. Accordingly, over the sampled transitions set $\mathcal{D}_{\mathcal{Y}_i}$, the efficiency of base learner function f_θ is measured by the following loss function:

$$\mathcal{L}(f_{\boldsymbol{\theta}}; \mathcal{D}_{\mathcal{Y}_i}) = \mathbb{E}_{\mathcal{D}_{\mathcal{Y}_i}} \left[(r + \gamma \max_{a' \in \mathcal{A}} Q(\boldsymbol{s}', a' | f_{\hat{\boldsymbol{\theta}}_{\mathcal{Y}_i}}) - Q(\boldsymbol{s}, a | f_{\boldsymbol{\theta}_{\mathcal{Y}_i}})) \boldsymbol{\phi}_{\mathcal{Y}_i}(\boldsymbol{s}, a) \right], \tag{5}$$

where $\hat{\theta}_{\mathcal{Y}_i}$ marks the parameters of target base learner that are fixed for every $\tilde{t}$ iterations.

Note that we did not follow the strict multivariate Fourier Series [9], but made adjustments to adapt it to the RL setting. First, $\sin$ terms of the Fourier series are dropped since the Fourier basis function are guaranteed to be even, throwing away the sin term by symmetric operations not only avoids influence the calculations, but also reduces the complexity of the evaluation [9]. Second, to address the credit assignment problem, the method imposes temporal-difference errors and an eligibility traces vector. The addition of these two items allows the statuses visited to be recorded in a discounted cumulative manner during the weight update process. Also, the method follows the experience replay scheme to maintain a experience memory, save the experience as transition $\boldsymbol{d}_i^t = (\boldsymbol{s}_i^t, a_i^t, r_i^t, \boldsymbol{s}_i^{t+1})$. The replay maintains a constant size, as training progresses, new transitions top off the old ones, agents sample mini-batches from memory in random to upgrade the policy. Finally, same in other works, the ϵ-greedy exploration strategy is adopted to address the exploration-exploitation dilemma: In addition to the typical selection of actions based on the highest Q-value guidance, the method retains a ϵ probability of choosing actions at random.

B. Model-agnostic Meta-Learner Framework

Our study aligns with the principles of the gradient-based model-agnostic meta-reinforcement learning framework MAML [4], [10], to find a well-generalized global initialization $\boldsymbol{\theta}_0$ of base learner f that can adapt to all traffic environments. Its pseudo code is given in Alg. 1.

Meta-Training: In the model's training phase, the model parameters' optimization is performed by gradient descent between the base learners and meta-learner alternately. Each gradient step requires a newly sampled transitions batch $\mathcal{D}$ with the current policy $f_{\boldsymbol{\theta}}$.

First, the method iteratively updates the parameters of the basic learner $f_{\boldsymbol{\theta}_{\mathcal{Y}_i}}$, $\boldsymbol{\theta}_{\mathcal{Y}_i} \in \{\boldsymbol{\theta}_{\mathcal{Y}_a}, \cdots, \boldsymbol{\theta}_{\mathcal{Y}_n}\}$ with the transitions batch $\mathcal{D}_{\mathcal{Y}_i}$ sampled from $\mathcal{Y}_i$ correspondingly, where update at each gradient step takes form as: $\boldsymbol{\theta}_{\mathcal{Y}_i} \leftarrow \boldsymbol{\theta}_{\mathcal{Y}_i} - \alpha \nabla_{\boldsymbol{\theta}} \mathcal{L}_{\mathcal{D}_{\mathcal{Y}_i}}(f_{\boldsymbol{\theta}})$, α are step size scalars, the loss function $\mathcal{L}$ using Eqn. 5 as previously defined.

Afterwards, after the adaptation of each basic learner $\mathcal{Y}_i$ is aggregated, meta-learner is learned by updating $\boldsymbol{\theta}_0$ using a

newly sampled transitions $\mathcal{D}'$: $\boldsymbol{\theta}_0 \leftarrow \boldsymbol{\theta}_0 - \beta \sum_{\mathcal{Y}_i} \nabla_{\boldsymbol{\theta}} \mathcal{L}_{\mathcal{D}'_{\mathcal{Y}_i}}(f_{\boldsymbol{\theta}})$, where β are step size hyperparameters.

When the training is over, there are a set well-generalized parameters $\boldsymbol{\theta}_0$ of f with universal adaptation. The loss function of the meta-training process takes the form as

$$\mathcal{L}(f_{\boldsymbol{\theta}_0}; \mathcal{D}'_{\mathcal{Y}_i}) = \mathbb{E}_{\mathcal{D}'_{\mathcal{Y}_i}} \left[f_{\boldsymbol{\theta}_0 - \alpha \nabla_{\boldsymbol{\theta}} \mathcal{L}(f_{\boldsymbol{\theta}}; \mathcal{D}_{\mathcal{Y}_i})} \right]. \tag{6}$$

Meta-Testing: To get the particular model f to the new traffic environment $\mathcal{Y}_g$, we put the above trained agents $\boldsymbol{\theta}_0$ as initialization to process update $\boldsymbol{\theta}_{\mathcal{Y}_g}$:

$$\boldsymbol{\theta}_{\mathcal{Y}_g} \leftarrow \boldsymbol{\theta}_{\mathcal{Y}_g} - \alpha \nabla_{\boldsymbol{\theta}} \mathcal{L}_{\mathcal{D}_{\mathcal{Y}_g}}(f_{\boldsymbol{\theta}}). \tag{7}$$

Then within a finite gradient steps, we can obtain a set of finally gained optimal $\boldsymbol{\theta}_{\mathcal{Y}_g}$.

V. Experiments

In this section, we validate the effectiveness and efficiency of our proposed MetaSignal by conducting series of comparative experiments on both synthetic and real-world datasets. First, the experimental setup and datasets are described. Subsequently, we give detailed experimental results and the corresponding analysis.

A. Environment Platform

We run experiments on CityFlow [28], an advanced simulation platform that offers the simulation environments for traffic signal control tasks. Based on the traffic flow dataset, the simulator pushes vehicles into the road network at their respective departure times, and simulates their travelling along predefined routes. By observing the traffic situation and implementing signal actions determined by the control method, Cityflow allows for the simulation of signal control for traffic environment.

B. Evaluation Criteria

We utilize the following two representative measures as evaluation criteria to digitally assess the performance of the different methods.

- *Average Travel time* measured in seconds, represents the duration a vehicle takes to cross the intersection, i.e., the time difference between entering and exiting. The smaller the value, the better the vehicle driving experience, thus reflecting better algorithm performance. This metric is extensively utilized in the literature.

- *Speed Score* indicates the ratio of vehicle speed through the intersection to the current maximum speed limit at the intersection. The index reflects the intensity at the intersection since slower speeds contribute to increased intensity. Larger index values signify reduced intensity at the intersection.

C. Comparison Methods

We use the following representative methods for comparison to assess the validity of the proposed method.

· **DQN** [2]: provides a form of the most basic RL control algorithm, observing vehicle positions in the road network as states and taking the waiting queue length as the reward.

TABLE I: Settings of real-world and synthetic roadnets.

Type	Dataset	# Intersection	Road segments (m)
Synthetic	Syn_3 × 3	9 = 3 × 3	300 × 300
	Syn_4 × 4	16 = 4 × 4	300 × 300
Real-world	Jinan	12 = 3 × 4	800 × 400
	Hangzhou	16 = 4 × 4	800 × 600

· **TOSFB** [11]: utilizes a true online SARSA(λ) algorithm with Fourier basis functions (TOS(λ)-FB) to facilitate the traffic signal agents' operations. Here it shares the same RL environment with DQN.

· **MetaLight** [5]: builds on FRAP [18] as the base learner and implements meta-learning in the MAML framework. It can and only can transferring knowledge between different intersections.

· **GeneraLight** [6]: actualizes meta-learning as guided by HSML [23], which assumes that different environments can be grouped into finite clusters. It has proven performance on generated datasets with distinct cluster divisions. For all the following experiments, we search for the optimal results corresponding to all scenarios for which the clusters are numbered from 1 to 7.

· **PressLight** [13]: adheres to the classical theory of Max Pressure, with the road pressure serving as the central element of the RL agent. It optimizes the policy by leveraging the pressure of intersections.

· **IPDALight** [15]: proposes a new concept *Intensity*, which optimizes the definition of *pressure* to capture more of the traffic dynamics. In addition, it differs from other algorithms in that the phase duration is adjustable, but this means that it accordingly increases a lot of interaction requirements for the agent and the environment.

In all scenarios, the episode length is fixed at 3600 seconds, while the interaction interval between the simulator and RL agent for all methods is set to 10 seconds. Given that the non-meta-learning TSC approaches are not specifically tailored for training and testing in various traffic environments, to be fair, we used the MAML combined version of these methods, and add a shared replay memory for each method of the different environments. For all algorithms, the meta-learner obtains a relatively stable algorithm structure after the base learner has learned 200 episodes. For all experiments, we report the mean of their three randomized trials.

D. Datasets

We conducted experiments on two synthetic setting, Syn_3 × 3 , Syn_4 × 4, and two real-world city settings, Jinan and Hangzhou [1]. Each traffic dataset comprises two components: (1) the traffic network dataset and (2) the traffic flow dataset. The former provides information about the traffic network, including signal phases, parameters of lanes and intersections. The information of the four used traffic network datasets is summarized in Table I. The latter encompasses vehicles' travel details, consisting of the entry time of each

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

TABLE II: Overall performance comparison on the set of Syn_3x3 datasets, where *Travel*, *Speed* are the shorted version for Average travel time and Speed score. The optimal and suboptimal values in each column are highlighted in bold.

Model	$D_{S3\times3\_300\_0.3}$		$D_{S3\times3\_300\_0.6}$		$D_{S3\times3\_500\_0.3}$		$D_{S3\times3\_500\_0.6}$		$D_{S3\times3\_700\_0.3}$		$D_{S3\times3\_700\_0.6}$	
	Travel	Speed	Travel	Speed	Travel	Speed	Travel	Speed	Travel	Speed	Travel	Speed
DQN	159.35	0.66	154.72	0.67	229.68	0.47	319.25	0.36	698.15	0.19	686.61	0.21
TOSFB	235.53	0.44	277.18	0.38	416.19	0.27	526.81	0.26	670.40	0.21	697.57	0.22
MetaLight	691.05	0.16	708.78	0.16	893.38	0.13	942.78	0.13	1104.30	0.11	1061.55	0.12
GeneraLight	653.38	0.17	667.91	0.18	794.29	0.15	859.68	0.15	1011.91	0.12	966.03	0.14
PressLight	**147.70**	**0.70**	**151.45**	**0.68**	**172.27**	**0.61**	**243.70**	**0.48**	506.76	0.26	540.95	0.29
IPDALight	278.94	0.43	316.39	0.41	264.88	0.44	296.37	0.42	**293.26**	**0.41**	**319.56**	**0.38**
MetaSignal	**143.01**	**0.72**	**145.34**	**0.71**	**141.62**	**0.73**	**142.15**	**0.73**	**144.75**	**0.72**	**144.64**	**0.72**

TABLE III: Overall performance comparison on the set of Jinan and Hangzhou datasets, where *Travel*, *Speed* are the shorted version for Average travel time and Speed score. The optimal and suboptimal values in each column are highlighted in bold.

Model	D_{jn}		$D_{jn_{2000}}$		$D_{jn_{2500}}$		D_{hz}		$D_{hz_{5734}}$		$D_{hz_{5816}}$	
	Travel	Speed	Travel	Speed	Travel	Speed	Travel	Speed	Travel	Speed	Travel	Speed
DQN	955.67	0.23	974.88	0.23	899.76	0.24	1167.57	0.21	1343.56	0.17	835.87	0.25
TOSFB	544.07	0.43	455.43	0.52	487.98	0.48	541.98	0.54	799.21	0.43	543.22	0.56
MetaLight	1304.05	0.14	1374.48	0.14	1270.03	0.15	1275.70	0.19	1414.36	0.15	897.61	0.21
GeneraLight	1189.69	0.17	1265.19	0.17	1168.57	0.17	1157.73	0.24	1313.88	0.20	847.00	0.27
PressLight	**280.45**	**0.81**	**284.89**	**0.81**	**273.96**	**0.81**	**318.88**	**0.89**	669.99	0.47	422.12	0.72
IPDALight	351.42	0.65	366.68	0.63	352.18	0.65	343.08	0.84	**399.23**	**0.74**	**364.03**	**0.80**
MetaSignal	**289.31**	**0.78**	**287.93**	**0.79**	**290.19**	**0.78**	**317.14**	**0.90**	**318.40**	**0.90**	**319.79**	**0.89**

TABLE IV: Transfer performance comparison on the set of Syn_4x4 datasets, where the base model is obtained by training on the D_{hz} dataset. *Travel*, *Speed*, *Waiting* are the shorted version for Average travel time, Speed score and Max waiting time. The optimal and suboptimal values in each column are highlighted in bold.

Model	$D_{S4\times4\_700\_0.3}$			$D_{S4\times4\_700\_0.6}$			$D_{S4\times4\_750\_0.3}$			$D_{S4\times4\_750\_0.6}$		
	Travel	Speed	Waiting	Travel	Speed	Waiting	Travel	Speed	Waiting	Travel	Speed	Waiting
DQN	1348.02	0.09	3175.20	1378.15	0.09	3559.37	1376.99	0.08	3414.27	1393.90	0.08	3413.27
TOSFB	804.22	0.20	1331.93	868.17	0.20	1813.53	865.80	0.19	924.13	890.75	0.20	1291.20
MetaLight	1447.76	0.07	3403.47	1436.99	0.08	3520.00	1469.99	0.07	3490.03	1464.31	0.07	3491.93
GeneraLight	1389.70	0.08	3488.70	1379.29	0.09	3554.10	1401.55	0.08	3402.73	1422.34	0.08	3540.47
PressLight	1100.97	0.14	3042.47	1040.62	0.16	2988.27	1091.29	0.14	3029.17	1069.65	0.16	2719.93
IPDALight	**343.95**	**0.84**	**537.40**	**345.17**	**0.84**	**451.83**	**359.08**	**0.80**	**799.73**	**343.62**	**0.84**	**568.47**
MetaSignal	**319.41**	**0.89**	**277.17**	**317.08**	**0.90**	**221.37**	**319.16**	**0.89**	**264.27**	**318.79**	**0.90**	**236.30**

vehicle into the traffic network and their pre-planned routes from the origin to the destination. The following are the details of the traffic flow datasets under varying road network settings:

· **Synthetic data:** Differentiated traffic configurations are generated to simulate diverse traffic demands based on the combination of various average vehicle arrival rates and distinct traffic distribution templates. With the Flat and Peak templates represented by variances of 0.3 and 0.6, respectively, the simulated traffic flows for both the Syn_3×3 and Syn_4×4 road networks are generated, applying arrival rates of 300, 500, and 700 (vehicles/h/lane) for the former and 700 and 750 (vehicles/h/lane) for the latter. As a result, there are 10 different sets of synthetic traffic flow datasets, and all vehicles access and exit the network through the rim edges.

· **Real-world data:** Each city has three different real traffic flow datasets. The following works label the different traffic flows in Jinan and Hangzhou with D_{jn}, $D_{jn_{2000}}$, $D_{jn_{2500}}$ and D_{hz}, $D_{hz_{5734}}$, $D_{hz_{5816}}$ accordingly.

E. Overall Performance

To test the meta-learning ability on synthetic datasets, we first conducted experiments on the series of datasets of the Syn_3x3 road network setting. All algorithms were trained based on the Syn_3x3_500_0.3 dataset separately, and the trained parameters were used as initial parameters for subsequent tasks for one round of meta-training, all the detailed performance test values are given in Table II. From the results, it is clear that indicator Average travel time and Speed score have synchronization. For simple scenarios such as $D_{S3\times3\_300\_0.3}$, baselines DQN, TOSFB, PressLight and IPDALight all reflect relatively excellent control performance. Besides, as shown by the results of baselines, along with the increase in transportation pressure represented by the dataset, the challenge accepted by the ATSC methods gets harder, and

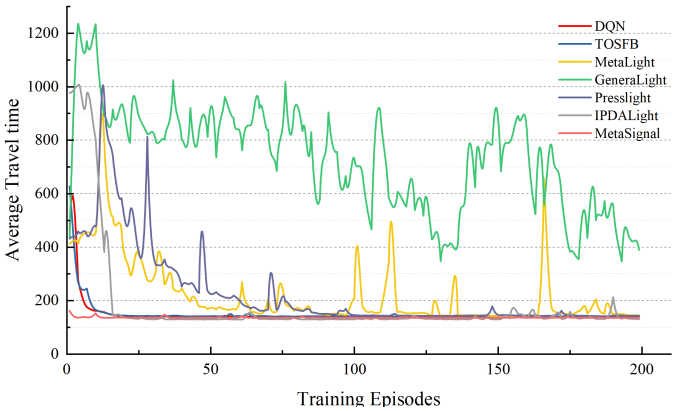
its results are unavoidably decreasing. Instead, our proposed MetaSignal ignores the traffic pressure demands embodied in the traffic flow datasets themselves, and is able to provide the best traffic experience in all scenarios through rational scheduling. Also, it is worth mentioning that for GeneraLight, its best results for the three datasets with variance of 0.3 are 5,6,6, while its best results for the three datasets with variance of 0.3 are all as 1, this is consistent with the logic of the algorithm.

Then, we further tested the meta-learning ability of all the algorithms on the two sets of real datasets by using D_{jn} and D_{hz} as the source training datasets, respectively, all the detailed performance test values are given in Table III. First, unfortunately, we can see that the performance of DQN as the representative of the base algorithm falls dramatically in the face of complex realistic scenarios. Meanwhile, the other base algorithm TOSFB uses linear approximation, which employs the same RL environment setting, performs nearly double better than DQN due to the nature of its error bounds. It is necessary to note that PressLight works best on the three datasets of the Jinan road network, meanwhile, our proposed MetaSignal exhibits the same good performance by a small gap. Besides, PressLight demonstrates poor control performance in dealing with the traffic flow data D_{hz}, which causes difficulties for all the baselines. The proposed method maintains the optimal performance on all the datasets configured in Hangzhou settings, demonstrating a stable and robust processing capability for different scenarios. As a reference, GeneralLight assumes that the corresponding clusters for the traffic flows corresponding to the Jinan and Hangzhou road networks are 2 and 3, respectively.

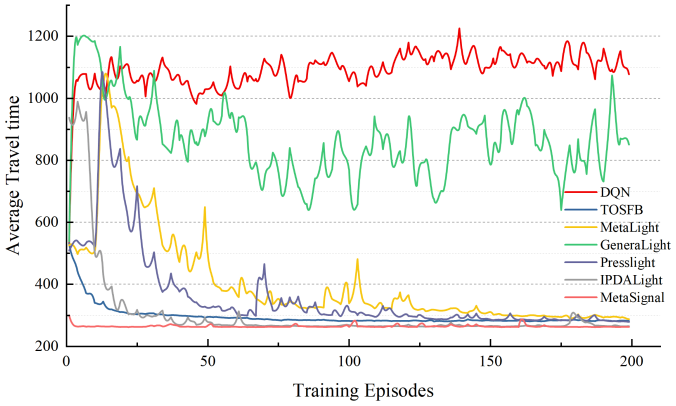
F. Transfer Learning Testing

In this subsection, we test the transfer ability of various methods by using the meta-learners obtained by the corresponding method trained on the D_{hz} real dataset as the initial models corresponding to four different configurations of traffic flow models on the Syn_4x4 road network. All the detailed performance test values are given in Table IV. It can be seen that all baselines, except TOSFB and IPDALight, have poor transfer learning capacity, where the performance of TOSFB is guaranteed by the error bound property of the linear approximation it employs. The performance of IPDALight in general maintains the control level corresponding to the real Hangzhou datasets, which reflects that considering more of the moving dynamic properties in RL modeling endows the algorithm with the robustness to face the transfer tasks. This fact is also reflected in the performance of the proposed MetaSignal for this task. As a reference, GeneralLight considers that these traffic flows correspond to clusters 3 or 4.

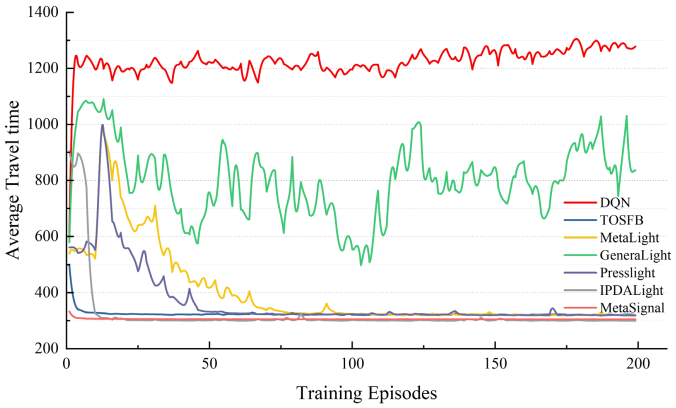
An additional indicator statistic, maximum waiting time, is also given in Table IV. It serves as a marker of how long a vehicle waits for a single red light to pass on the intersection in the current simulation process, again is a feedback of the vehicle's driving experience, the larger its value, the worse the experience. As can be seen from IPDALight's tests on $D_{S4\times4\_700\_0.3}$, this metric does not strictly correlate with the other two indicators. Although the driving experience provided by IPDALight does not differ much from the optimal performance represented by our algorithm in terms of average traveling time, the maximum waiting time for vehicles on its road network is approximately three times longer than the corresponding value of our proposed MetaSignal, signifying a compromise in the vehicle experience. Conversely, the proposed method considers all aspects in a more balanced manner, providing a fairer control logic.



(a) Syn_3x3_500_0.3



(b) Jinan



(c) Hangzhou

Fig. 2: Average travel time along episodes for (a)Syn_3x3_500_0.3, (b)D_{jn}, (c)D_{hz} datasets.

G. Analysis of Training Curves

In this subsection, we give the training curves of all the algorithms on the three datasets $D_{S3\times3\_500\_0.3}$, D_{jn}, D_{hz} in turn in Fig. 2 to provide a reference on the learning ability of the base learners of all the methods. Most obviously, GeneralLight performs fluctuating and difficult to converge on all tasks due to its requirement to find the target cluster value among the circled number of clusters. Besides, although the DQN showed good convergence on the first task, it quickly fell into the quagmire of exploding training metrics on the subsequent tasks, which is consistent with the test results in the above mentioned scenarios. It is also worth mentioning that although MetaLight demonstrated instability during the training of the first task, it showed favorable adaptation in the two subsequent tasks, which unfortunately was not matched in the test results. Furthermore, although most algorithms eventually converge to nearly identical optimal solutions, all neural network-based implementations need to undergo a period of dramatic fluctuations in which the worst values are unacceptable. Meanwhile, the algorithms TOSFB, our method implemented based on linear approximation both start with reasonable values and converge quickly to the optimal values, maintain their stability throughout the subsequent training process.

VI. Conclusion

Reinforcement-based learning has demonstrated significant benefits for traffic signal control, owing to its ability to receive feedback from environmental interactions. However, due to the need for these interactions, applying the RL algorithm to completely new scenarios necessitates a lengthy training period. In this paper, we introduce MetaSignal, a novel traffic signal control framework that utilizes meta-learning to facilitate knowledge transfer, enhancing the learning efficiency of reinforcement learning in new scenarios. Specifically, we employ the Fourier basis function for linearly approximating the value function. The selected linear approximation guarantees convergence and provides verifiable error bounds, distinguishing it from other methods such as neural networks. This property enhances the robustness of the learned knowledge, facilitating its applicability across diverse traffic scenarios. The conducted experiments demonstrate the capability of our model to handle diverse traffic scenarios, achieving superior performance and high efficiency.

For future work, we want to validate the method's performance in a wider range of traffic environments, including irregular road networks or larger environments with multiple intersections. Additionally, the current traffic conditions on the network are simplified, disregarding factors such as the waiting period represented by the yellow light or the needs of non-motorized vehicles and pedestrians, which are also areas of concern. Furthermore, the linear approximation function used in the model comprises fewer parameters, offering a viable approach to demystifying the black-box nature of reinforcement learning and providing a rational explanation for the model's behavior.

Acknowledgment

This work is partially supported by JSPS KAKENHI Grant Numbers JP20H04174, JP22K11989, Leading Initiative for Excellent Young Researchers (LEADER), MEXT, Japan, and JST, PRESTO Grant Number JPMJPR21P3, Japan. Shuning Huang is the corresponding author.

References

[1] A. L. Bazzan and F. Klügl, *Introduction to intelligent systems in traffic and transportation*. Springer Nature, 2022.

[2] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski *et al.*, "Human-level control through deep reinforcement learning," *nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[3] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.

[4] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *Proceedings of the 34th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, D. Precup and Y. W. Teh, Eds., vol. 70. PMLR, 06–11 Aug 2017, pp. 1126–1135.

[5] X. Zang, H. Yao, G. Zheng, N. Xu, K. Xu, and Z. Li, "Metalight: Value-based meta-reinforcement learning for traffic signal control," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 01, pp. 1153–1160, Apr. 2020.

[6] H. Zhang, C. Liu, W. Zhang, G. Zheng, and Y. Yu, "Generalight: Improving environment generalization of traffic signal control via meta reinforcement learning," ser. CIKM '20. New York, NY, USA: Association for Computing Machinery, 2020, p. 1783–1792.

[7] K. Li, Z. Liu, N. Ding, and R. Jiang, "A dqn algorithm for traffic signal control based on meta-learning training," in *Proceedings of the 2023 7th International Conference on Digital Signal Processing*, ser. ICDSP '23. New York, NY, USA: Association for Computing Machinery, 2023, p. 64–70.

[8] J. Tsitsiklis and B. Van Roy, "An analysis of temporal-difference learning with function approximation," *IEEE Transactions on Automatic Control*, vol. 42, no. 5, pp. 674–690, 1997.

[9] G. Konidaris, S. Osentoski, and P. Thomas, "Value function approximation in reinforcement learning using the fourier basis," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 25, no. 1, pp. 380–385, Aug. 2011.

[10] R. Vuorio, S.-H. Sun, H. Hu, and J. J. Lim, "Toward multimodal model-agnostic meta-learning," 2018.

[11] L. N. Alegre, T. Ziemke, and A. L. C. Bazzan, "Using reinforcement learning to control traffic signals in a real-world scenario: An approach based on linear function approximation," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 7, pp. 9126–9135, 2022.

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

[13] H. Wei, C. Chen, G. Zheng, K. Wu, V. Gayah, K. Xu, and Z. Li, "Presslight: Learning max pressure control to coordinate traffic signals in arterial network," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, ser. KDD '19. New York, NY, USA: Association for Computing Machinery, 2019, p. 1290–1298.

[14] C. Chen, H. Wei, N. Xu, G. Zheng, M. Yang, Y. Xiong, K. Xu, and Z. Li, "Toward a thousand lights: Decentralized deep reinforcement learning for large-scale traffic signal control," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 04, pp. 3414–3421, Apr. 2020.

[15] W. Zhao, Y. Ye, J. Ding, T. Wang, T. Wei, and M. Chen, "Ipdalight: Intensity- and phase duration-aware traffic signal control based on reinforcement learning," *Journal of Systems Architecture*, vol. 123, p. 102374, 2022.

[16] V. H. Pham and H.-S. Ahn, "Distributed stochastic mpc traffic signal control for urban networks," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–18, 2023.

[17] R. Dai, C. Ding, X. Wu, B. Yu, and G. Lu, "Coupling control of traffic signal and entry lane at isolated intersections under the mixed-autonomy traffic environment," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–15, 2023.

[18] G. Zheng, Y. Xiong, X. Zang, J. Feng, H. Wei, H. Zhang, Y. Li, K. Xu, and Z. Li, "Learning phase competition for traffic signal control," in *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, ser. CIKM '19. New York, NY, USA: Association for Computing Machinery, 2019, p. 1963–1972.

[19] M. Chen, T. Wang, K. Ota, M. Dong, M. Zhao, and A. Liu, "Intelligent resource allocation management for vehicles network: An a3c learning approach," *Computer Communications*, vol. 151, pp. 485–494, 2020.

[20] C. Zhang, M. Dong, and K. Ota, "Employ ai to improve ai services : Q-learning based holistic traffic control for distributed co-inference in deep learning," *IEEE Transactions on Services Computing*, vol. 15, no. 2, pp. 627–639, 2022.

[21] Z. Ying, S. Cao, X. Liu, Z. Ma, J. Ma, and R. H. Deng, "Privacysignal: Privacy-preserving traffic signal control for intelligent transportation system," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 9, pp. 16 290–16 303, 2022.

[22] W. Du, J. Ye, J. Gu, J. Li, H. Wei, and G. Wang, "Safelight: A reinforcement learning method toward collision-free traffic signal control," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, no. 12, pp. 14 801–14 810, Jun. 2023.

[23] H. Yao, Y. Wei, J. Huang, and Z. Li, "Hierarchically structured meta-learning," in *Proceedings of the 36th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, K. Chaudhuri and R. Salakhutdinov, Eds., vol. 97. PMLR, 09–15 Jun 2019, pp. 7045–7054.

[24] J. Wu, M. Dong, K. Ota, J. Li, and Z. Guan, "Big data analysis-based secure cluster management for optimized control plane in software-defined networks," *IEEE Transactions on Network and Service Management*, vol. 15, no. 1, pp. 27–38, March 2018.

[25] H. Seijen and R. Sutton, "True online td(lambda)," in *Proceedings of the 31st International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, E. P. Xing and T. Jebara, Eds., vol. 32, no. 1. Bejing, China: PMLR, 22–24 Jun 2014, pp. 692–700.

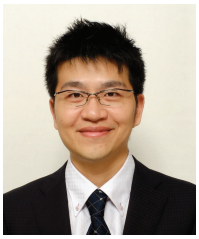
[26] H. van Seijen, A. R. Mahmood, P. M. Pilarski, M. C. Machado, and R. S. Sutton, "True online temporal-difference learning," *Journal of Machine Learning Research*, vol. 17, no. 145, pp. 1–40, 2016.

[27] Y. Li, G. Chen, and Y. Zhang, "Cycle-based signal timing with traffic flow prediction for dynamic environment," *Physica A: Statistical Mechanics and its Applications*, vol. 623, p. 128877, 2023.

[28] Z. Tang, M. Naphade, M.-Y. Liu, X. Yang, S. Birchfield, S. Wang, R. Kumar, D. Anastasiu, and J.-N. Hwang, "Cityflow: A city-scale benchmark for multi-target multi-camera vehicle tracking and re-identification," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

Shuning Huang received the B.S. degree in Information and Computing Science from Taiyuan University of Technology, Taiyuan, China, in 2016, and the M.S. degree in School of Computer Science & Technology from Soochow University, Suzhou, China, in 2019. She is currently pursuing the Ph.D. degree in Electrical Engineering at Muroran Institute of Technology, Japan. Her main fields of research interest include recommendation and artificial Intelligence.

Kaoru Ota was born in Aizu-Wakamatsu, Japan. She received M.S. degree in Computer Science from Oklahoma State University, the USA in 2008, B.S. and Ph.D. degrees in Computer Science and Engineering from The University of Aizu, Japan in 2006, 2012, respectively. Kaoru is a Professor and Ministry of Education, Culture, Sports, Science and Technology (MEXT) Excellent Young Researcher with the Department of Sciences and Informatics. She is also the founding Director of Center for Computer Science (CCS) at Muroran Institute of Technology, Japan. From March 2010 to March 2011, she was a visiting scholar at the University of Waterloo, Canada. Also, she was a Japan Society of the Promotion of Science (JSPS) research fellow at Tohoku University, Japan from April 2012 to April 2013. Kaoru is the recipient of IEEE TCSC Early Career Award 2017, The 13th IEEE ComSoc Asia-Pacific Young Researcher Award 2018, 2020 N2Women: Rising Stars in Computer Networking and Communications, 2020 KDDI Foundation Encouragement Award, and 2021 IEEE Sapporo Young Professionals Best Researcher Award, The Young Scientists' Award from MEXT in 2023. She is Clarivate Analytics 2019, 2021, 2022 Highly Cited Researcher (Web of Science) and is selected as JST-PRESTO researcher in 2021, Fellow of EAJ in 2022.



Mianxiong Dong received B.S., M.S. and Ph.D. in Computer Science and Engineering from The University of Aizu, Japan. He is the Vice President and Professor of Muroran Institute of Technology, Japan. He was a JSPS Research Fellow with School of Computer Science and Engineering, The University of Aizu, Japan and was a visiting scholar with BBCR group at the University of Waterloo, Canada supported by JSPS Excellent Young Researcher Overseas Visit Program from April 2010 to August 2011. Dr. Dong was selected as a Foreigner Research Fellow (a total of 3 recipients all over Japan) by NEC C&C Foundation in 2011. He is the recipient of The 12th IEEE ComSoc Asia-Pacific Young Researcher Award 2017, Funai Research Award 2018, NISTEP Researcher 2018 (one of only 11 people in Japan) in recognition of significant contributions in science and technology, The Young Scientists' Award from MEXT in 2021, SUEMATSU-Yasuharu Award from IEICE in 2021, IEEE TCSC Middle Career Award in 2021. He is Clarivate Analytics 2019, 2021, 2022 Highly Cited Researcher (Web of Science) and Foreign Fellow of EAJ.

Huan Zhou received his Ph. D. degree from the Department of Control Science and Engineering at Zhejiang University. He was a visiting scholar at the Temple University from Nov. 2012 to May, 2013, and a CSC supported postdoc fellow at the University of British Columbia from Nov. 2016 to Nov. 2017. Currently, he is a full professor in the College of Computer and Information Technology, China Three Gorges University. He was a Lead Guest Editor of Pervasive and Mobile Computing, and Special Session Chair of the 3rd International Conference on Internet of Vehicles (IOV 2016), and TPC member of IEEE WCSP'13'14, CCNC'14'15, ICNC'14'15, ANT'15'16, IEEE Globecom'17'18, ICC'18'19, etc. He has published more than 50 research papers in some international journals and conferences, including IEEE JSAC, TPDS, TWC and so on. His research interests include Mobile Social Networks, VANETs, Opportunistic Mobile Networks, and mobile data ofoading. He receives the Best Paper Award of I-SPAN 2014 and I-SPAN 2018, and is currently serving as an associate editor for IEEE ACCESS and EURASIP Journal on Wireless Communications and Networking.