2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 1-5, 2023. Detroit, USA

Effective Traffic Signal Control with Offline-to-Online Reinforcement Learning

Jinming Ma and Feng Wu

Abstract— **Reinforcement learning (RL) has emerged as a promising approach for optimizing traffic signal control (TSC) to ensure the efficient operation of transportation networks. However, the traditional trial-and-error technique in RL is usually impractical in real-world applications. Offline RL, which trains models using pre-collected datasets, is a more practical approach. However, this presents challenges such as suboptimal datasets and limited generalization of pre-trained models. To address this, we propose an offline-to-online RL framework for TSC that pre-trains a generalized model and quickly adapts to new traffic scenarios through online refinement. In the offline stage, we augment the pre-collected datasets to cover a diverse set of possible scenarios and use an offline RL method to pre-train a control model. To ensure generalization, we use FRAP-like network as our base model, which is designed to learn the basic logic for signal control. In the online stage, we introduce a discrepancy measure to tackle inconsistencies between offline pre-trained models and online scenarios and prioritize samples based on it. In the experiments, the proposed approach achieves competitive performance and reduces the training time needed for learning in new scenarios, compared to several baselines.**

I. INTRODUCTION

In recent years, reinforcement learning (RL) [1]–[4] has emerged as a promising approach to optimizing traffic signal control (TSC), which plays a critical role in ensuring the efficient operation of transportation networks. By learning from the interactions with the environment, RL agents can optimize TSC that adapts to real-time traffic patterns. However, to achieve well-trained models, traditional RL methods usually require a large number of interactions with the real system. This can be prohibitively expensive and may lead to severe traffic congestion or accidents. Therefore, most of the existing methods in the literature are difficult to deploy, given that trial-and-error techniques of RL are not practical in real-world applications.

A more practical approach is to employ offline RL [5]–[10] to train models using pre-collected datasets. In our daily life, it is often feasible to collect large amounts of traffic data with existing traffic infrastructure, such as road cameras and sensors. Then, the dataset can be pre-processed and transformed into a format suitable for offline RL training. However, there are still several challenges that need to be addressed. Firstly, the dataset used to train the policy may be *suboptimal*, leading to poor performance of the trained agent.

This work was supported in part by the Major Research Plan of the National Natural Science Foundation of China (Grant No. 92048301) and Anhui Provincial Natural Science Foundation (Grant No. 2208085MF172).

Both authors are with the School of Computer Science and Technology, University of Science and Technology of China, Hefei, Anhui, China (E-mail: `jinmingm@mail.ustc.edu.cn`, `wufeng02@ustc.edu.cn`).
Feng Wu is the corresponding author.

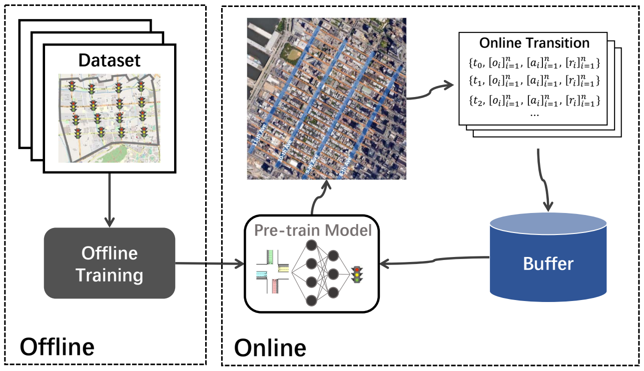

Dataset
Offline Training
Offline
Online Transition
$\{t_0, [o_i]_{i=1}^n, [a_i]_{i=1}^n, [r_i]_{i=1}^n\}$
$\{t_1, [o_i]_{i=1}^n, [a_i]_{i=1}^n, [r_i]_{i=1}^n\}$
$\{t_2, [o_i]_{i=1}^n, [a_i]_{i=1}^n, [r_i]_{i=1}^n\}$
...
Pre-train Model
Buffer
Online

Fig. 1. Illustration of our framework. In the offline stage, we employ offline reinforcement learning to train a pre-trained model on the static dataset. In the online stage, the pre-trained model is refined using online transitions obtained by interacting with the environment.

Secondly, the *generalization* of the trained policy is usually very limited, as it may not perform well under different traffic flows or network configurations. To mitigate these challenges, an online fine-tuning procedure is required to allow the agent to adapt to the new environment by gathering additional experiences. The fine-tuning process enables the agent to learn from real feedback and refine its policy, leading to improved performance.

Against this background, we propose a novel offline-to-online RL framework for TSC, as shown in Fig. 1, which allows for pre-training a generalized model and can rapidly adapt to new traffic scenarios. In the offline stage, we use an offline RL algorithm to pre-train a signal control model with the pre-collected dataset. Since the offline dataset is usually insufficient to capture the entire model, we augment the pre-collected datasets through several feasible solutions, such that a diverse set of possible scenarios are covered. For example, traffic flows in opposite directions in the morning and evening due to commuting. As different intersections may exhibit different traffic patterns and structures, it is important to adopt a flexible approach that accommodates these variances. To ensure the generalizability of pre-trained models in accounting for various traffic flows and intersection structures, we implement the network structure inspired by FRAP [11]. This network is specifically designed to learn the basic logic for signal control, i.e., phase competition, regardless of the intersection structure and the local traffic situation. By doing this, we can develop a well-generalized initialization from pre-collected datasets. Given this, our method can be effectively applied to various traffic conditions

2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) | 978-1-6654-9190-7/23/$31.00 ©2023 IEEE | DOI: 10.1109/IROS55552.2023.10341718

and intersection layouts.

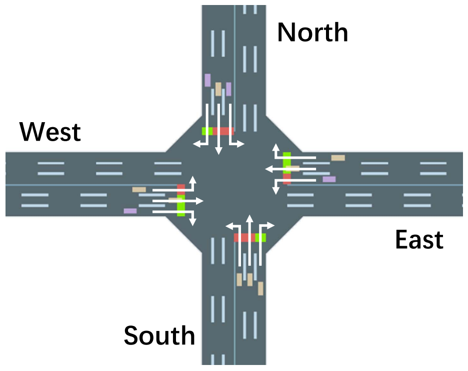
Due to the inconsistency between the offline pre-trained models and the online scenarios, the pre-trained model may not achieve optimal performance in new instances. To address this, an online fine-tuning process is used to adapt the agent to the new scenarios with online interactions. Since online interaction is expensive, we expect the pre-trained model can be fine-tuned with fewer online interactions and quickly adapt to new traffic scenarios. In order to achieve such efficiency, we introduce a discrepancy measure based on the inconsistency between the offline pre-trained models and the online scenarios, then prioritize samples based on it. By adjusting the sampling distribution for policy learning, the policy evaluation can be carried out more accurately and closer to the new scenarios. As a result, the performance of the learned policy is improved.

To the best of our knowledge, we are the first to propose an offline-to-online learning paradigm for RL-based TSC. Our experiments show that the proposed framework achieves competitive performance and effectively reduces the training time and computational resources compared with learning in new traffic scenarios from scratch.

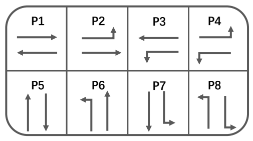
II. Related Works

We briefly review some of the most relevant approaches, including conventional methods and deep RL methods. A classic conventional approach for TSC is SOTL [12], which is controlled with demand-responsive rules comparing the current phase with current traffic. Another conventional method is MaxPressure [13], which aims to control the intersection by balancing queue length between neighboring intersections by minimizing the "pressure" of the phases. It is considered the state-of-the-art (SOTA) method for network-level TSC. Rule-based methods rely on predefined sets of rules to control traffic signals, which are simple and efficient but cannot handle complex traffic scenarios.

In contrast, RL methods can learn to optimize TSC policies through interactions with the environment. For example, [1] proposes an intelligent traffic signal control system based on deep Q-network (DQN). PressLight [14] uses the MaxPressure theory and designs the pressure as the reward of the agents, which has good performance in multi-intersection TSC. MA2C [15], [16] uses multi-agent advantage actor-critic (A2C) to cooperatively control multi-intersections. It includes information about neighborhoods and spatial discount to stabilize the training. MetaLight [3] utilizes meta-learning to adapt to the dynamics of the environment. However, they require a large number of training episodes, which can be time-consuming and costly. Furthermore, trial-and-error learning through interaction with the environment is not feasible in practice. Therefore, we focus on offline RL methods that learn from offline experience data [17]. It can significantly reduce training time and computational resources, making it more practical for real-world applications.



(a) Intersection and its 12 movements.



(b) 8 Phases.

Fig. 2. Illustration of the intersection definitions.

III. Background

Here, we focus on TSC with multiple intersections in a road network. As illustrated in Figure 2, we describe the terms that are commonly used in the research literature [18].

- **Lane**: A typical intersection usually contains roads in four directions: north, south, west, and east respectively. Correspondingly, each road has three types of lanes: left-lane, through-lane, and right-lane.
- **Traffic movement**: A traffic movement is defined as the movement of traffic in a particular direction, such as a left turn, through, or right turn. As depicted in Figure 2(a), the signal controls eight traffic movements, with the right turn traffic exempted from signal control but required to yield on a red light in accordance with traffic regulations in most countries.
- **Movement signal**: A movement signal is defined as a signal regulating a traffic movement, with green indicating the movement is permitted and red indicating it is prohibited.
- **Phase**: A phase is defined as a combination of the movement signals. It should be noted that some signals cannot turn "green" at the same time, i.e., conflicting signals. Note that the combination of the movement signals must be non-conflicting signals.

The objective of TSC is to develop an optimal phase cycle to enhance traffic conditions that minimize lane queue length and the average time spent by vehicles on approaching lanes.

A. *Markov Game*

TSC with multiple intersections is usually modeled as a Markov game, where each intersection in the traffic network is controlled by an agent. Formally, Markov game with N agents can be defined as a tuple $\langle \mathscr{S}, \{\mathscr{A}\}_{i=1}^{N}, \{\mathscr{O}\}_{i=1}^{N}, \mathscr{P}, \mathscr{R}, \gamma \rangle$. Here, $s \in \mathscr{S}$ denotes the true state of the environment. Each agent i chooses an action $a_i \in \mathscr{A}_i$ at each time step, forming a joint action vector $\mathbf{a} = [a_i]_{i=1}^{N}$. Different from the MDP, the next state follows the transition function $\mathscr{P}(s'|s,\mathbf{a}) : \mathscr{S} \times \mathscr{A}^N \times \mathscr{S} \to [0,1]$. Meanwhile, each agent i receives a local (partial) observation $o_i \in \mathscr{O}_i$ of the state and a reward r based on the shared reward function $\mathscr{R} : S \times \mathscr{A}^N \to \mathbb{R}$. The goal is to find a set of optimal policies $\pi = \{\pi_1, ..., \pi_N\}$, where each agent aims to maximize its own discounted return $\sum_{t=0}^{T} \gamma^t r_i^t$.

B. Offline Dataset

We collect road information and control signals from the real traffic environment and structured them as the transition $\langle s, a, r, s' \rangle$ to build offline datasets $\mathscr{D}$. Considering each intersection is individually controlled by an agent, each agent i access the offline dataset D_i, which is collected locally by the sensor in intersection i. Then, we will detail the settings of $D_i = \{\langle o, a, r, o' \rangle\}$ for TSC:

- **Observation** $o \in O$**:** Each agent observes part of the system state as its own observation. The observed traffic information is some quantitative descriptions of the intersection i, i.e., queue length, waiting time, and delay. In this paper, the observation includes the current phase and the number of vehicles in every incoming lane.
- **Action** $a \in A$**:** For each agent, it decides which phase to be selected from a phase set. In other words, an action is an index of the phase.
- **Reward** $r \in \Re$**:** it denotes immediate rewards received by the agent for a transition from (s,a) to s'. Similar to the PressLight [14], the reward is defined as $r_i = -P_i$, where P_i is the pressure of intersection i. The pressure of an intersection i is defined as the sum of the absolute pressures of overall traffic movements.

Given the dataset D_i, each agent i learns a signal control policy $\pi : O \to A$ to optimize the traffic in the intersection. The policy π can be deployed to the real traffic system since it is learned from real data.

IV. METHOD

We present a novel approach for TSC based on offline-to-online RL consisting of two stages. In the offline stage, we pre-train a generalized model for TSC with data augmentations and new network architecture. In the online stage, we quantify the distribution shift between the offline datasets and the online environment, then correct it with the priority sample selection. More details will be described next.

A. Pre-training Model with Offline Dataset

In the offline training stage, we pre-train a generalized control model using the collected offline dataset $\mathscr{D}$. Unlike traditional RL, an agent cannot collect new data samples from the environment. Hence, we need several techniques to increase the generality of the control model. To this end, we approach the problem from two perspectives: the dataset and the network model.

1) Data Augmentation: In machine learning, data augmentation (DA) is a well-known technique that can enable local exploration using trajectories in the dataset [19]–[21]. We can generate new samples $\langle \hat{s}_t, \hat{a}_t, \hat{r}_t, \hat{s}_{t+1} \rangle$ from existing transition $\langle s_t, a_t, r_t, s_{t+1} \rangle$. Note that the transformed transition $\langle \hat{s}_t, \hat{a}_t, \hat{r}_t, \hat{s}_{t+1} \rangle$ needs to conform to the physical laws in the environment.

However, if not done properly, DA can have a detrimental effect on the offline RL process [22]. The main challenge with offline RL is the inability of the agent to interact with the environment to refine its policy through trial-and-error. Consequently, augmenting the training dataset with unrealistic or implausible samples may not only impede learning but also compromise the accuracy of the value functions computed by the Bellman backups. Therefore, it is essential to ensure that the augmentations preserve the validity and feasibility of the transitions. Fortunately, we can use transportation knowledge to verify the correctness of new samples.

In TSC, DA aims to simulate the variability of traffic patterns in real-world scenarios and generate new data samples for training the offline RL model. Here, we consider several strategies for DA in offline RL for TSC:

1. **Flipping**: flipping the intersection horizontally or vertically can create new training examples for a given intersection and can be helpful in training the model to recognize vehicles coming from different directions.
2. **Rotation**: rotating the intersection by a certain degree, i.e., 90°, 180° and 270° can help the model recognize different orientations of vehicles.
3. **Noise**: noise injection can also be used as a DA technique, where small amounts of noise are added to the state. This technique aims to simulate the stochasticity of real-world environments and improve the robustness of the trained model. However, such DA strategy also needs to coincide with the reward obtained from the augmented state-action pairs, i.e., $r(\hat{s}_t, \hat{a}_t)$ is reasonable. Fortunately, in the case of TSC, the reward calculation is based on heuristics, which offers some flexibility for incorporating augmented state-action pairs.

To put together, the DA provides a practical and efficient way to generate diverse and realistic data for training the offline RL model in TSC.

2) Network Model: When offline data is collected from various traffic scenarios, it is challenging to account for differences in intersection structures and phases in subsequent test scenarios. To address this, we borrow the FRAP [11] network for our offline-to-online TSC framework. In more detail, FRAP is a specially designed network architecture that learns the phase competition relationships in TSC, which can effectively share the learned knowledge with similar control logic. The property is especially helpful when tackling offline-to-online TSC.

Given this, we employ state-of-the-art offline algorithms like BCQ [5] to pre-train a generalized model using the offline dataset $\mathscr{D}$. Similar to single-agent offline RL, we assume that each agent i learns from its dataset $D_i \in \mathscr{D}$

collected by the corresponding traffic light, which contains transitions $\langle o_i, a_i, o_i', r\rangle$. The straightforward idea is to apply single-agent offline RL directly to each agent and learn a set of policies, one by one. The general form of optimization function for BCQ is as:

$$\begin{aligned}\mathscr{L}_\theta &= \mathbb{E}_{(o,a,o')\sim D_i}\left[r+\gamma Q_{\theta'}(o',a') - Q_\theta(o,a)\right],\\ \text{s.t.} \quad a' &= \underset{a'|G_w(a'|o')/max_{\hat{a}}G_w(\hat{a}|o')>\tau}{\arg\max} Q_\theta(o',a')\end{aligned} \tag{1}$$

where $G_w(a|o) \approx \pi_{D_i}(a|o)$ is the state-conditioned generative model to model the distribution of the dataset. The threshold τ maintains the batch-constrained in the dataset, where setting $\tau = 0$ returns Q-learning and $\tau = 1$ returns behavioral cloning. The objective enables us to train the Q-function by the offline dataset D_i, and the trained Q-function can be leveraged to infer the policy by the following:

$$\pi(o) = \underset{a|G_w(a|o)/max_{\hat{a}}G_w(\hat{a}|o)>\tau}{\arg\max} Q_\theta(o,a) \tag{2}$$

Note that the pre-trained model may not achieve optimal performance in a new environment. To address this, an online fine-tuning process is used to adapt the agent to the new scenarios by collecting additional experiences.

B. *Refining Model with Online Interactions*

Here, we aim to explore how to effectively refine the pre-trained model to facilitate knowledge transfer across different scenarios. However, the performance of online refinement is often limited due to the mismatch between various intersection structures and phase configurations. Therefore, we design a scheme that detects and rectifies the inconsistencies between the offline pre-trained models and online scenarios. For convenience, we denote the MDP of the offline dataset as $\mathscr{M}$ and the MDP of the new environment as $\hat{\mathscr{M}}$.

With the aim of detecting inconsistencies in new scenarios and utilizing new samples, we propose a metric to quantify such inconsistencies and prevent subsequent policy learning from collapsing due to differences in Q-value estimation. Specifically, we define the "inconsistency" as the difference in Q-value estimation between the pre-trained models and the online scenarios:

$$\mathscr{E} = \mathbb{E}_{(o,a)\sim\hat{\mathscr{M}}}\left[(Q_{\mathscr{M}}(o,a) - \tilde{Q}_{\hat{\mathscr{M}}}(o,a))^2\right] \tag{3}$$

where $Q_{\mathscr{M}}$ is the initial pre-trained model and $\tilde{Q}_{\hat{\mathscr{M}}} = r(o,a) + \gamma \max Q_{\mathscr{M}}(o',a')$ is the estimated value of the new environment. Note that $Q_{\mathscr{M}}$ is the pre-trained model learned from the offline dataset. Using the estimated Q-value difference as a metric enables us to detect and fix inconsistencies at the outset, thus facilitating the exploration of new environments.

Considering the diversity of dynamic transitions, we believe that those controversial data should be emphasized. Inspired by Prioritized Experience Replay (PER) [23], we store new transitions into the replay buffer with priority according to the inconsistency. Concretely, we define the probability of sampling transition i as:

$$P_i = \frac{\mathscr{E}_i^\alpha}{\sum_k \mathscr{E}_k^\alpha} \tag{4}$$

where the denominator item is all transitions in the buffer andα determines how much prioritization is used. By prioritizing samples with inconsistency, we can improve the generalization ability of the finetuned model to diverse traffic scenarios. Now, we update the Q-function by minimizing the following objectives:

$$\mathscr{L}_\theta = \mathbb{E}_{(o,a,o')\sim\mathscr{B}}\left[r+\gamma \max Q_{\theta'}(o',a') - Q_\theta(o,a)\right] \tag{5}$$

where $\mathscr{B}$ is the replay buffer and θ' is the target network.

V. Experiments

We empirically evaluate the effectiveness and efficiency of our approach as follows. Firstly, we assess the performance of our pre-training procedure by evaluating the learned policy on the corresponding road networks. Secondly, we demonstrate the generalizability of the pre-trained policy by refining it with online interactions, either in the original scenario or new scenarios. Finally, we conduct ablation studies to better understand the benefits of our key techniques.

A. *Experiments Setting*

Dataset. We conducted our experiments in three real-world traffic roadmap[1]: Jinan(JN), Hangzhou(HZ) and Manhattan(MAN). The road network settings are as follows:

- D_{HZ}: The road network of Hangzhou contains 16 intersections in a 4×4 city network. The traffic flow is generated from surveillance camera data.
- D_{JN}: The road network of Jinan contains 12 intersections in a 4×3 city network. The traffic flow is generated from surveillance camera data.
- D_{MAN}: The road network of Manhattan contains 48 intersections in a 16×3 city network. The number of vehicles generated is sampled from taxi trajectory data.

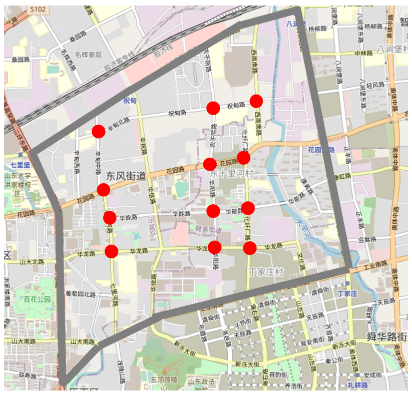
Among three real-world traffic roadmaps, we use *Hangzhou* with light traffic flow as the training scenarios. The testing scenarios are classified into two types and introduced as follows: 1) Homogeneous: the testing cities are similar to training cities except for traffic flow. 2) Heterogeneous: the testing cities and traffic flow are different from training cities. The detail of the train and test scenarios is summarized in Table I.

For offline training, we used a traditional TSC method — MaxPressure [13] to generate 20 trajectories with noise as the datasets. Since most real-world TSC applications currently use traditional methods, such datasets are considered more realistic and can be obtained from real-world applications.

Evaluation Metric. We used the *average travel time* (ATT) of all vehicles in a road network as the measure to evaluate different methods, which is the most frequently used

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

(a) Gudang Sub-district, Hangzhou, China


(b) Dongfeng Sub-district, Jinan, China

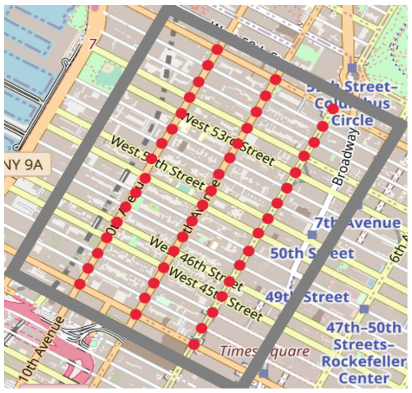
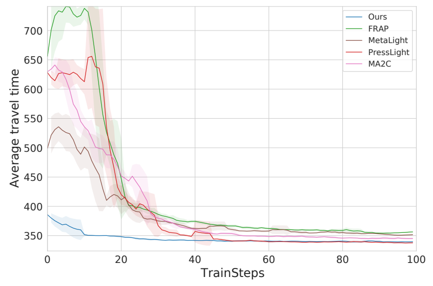
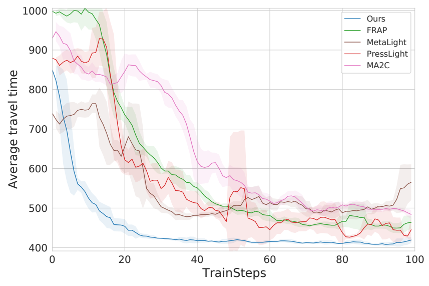
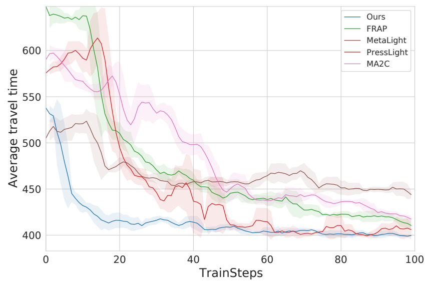
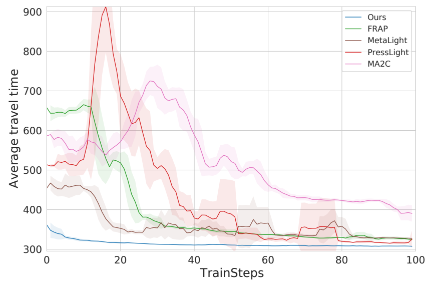

(c) Manhattan, New York City, USA

Fig. 3. Illustration of traffic networks in the real-world datasets.


(a) HZ1


(b) HZ2


(c) HZ3


(d) JN

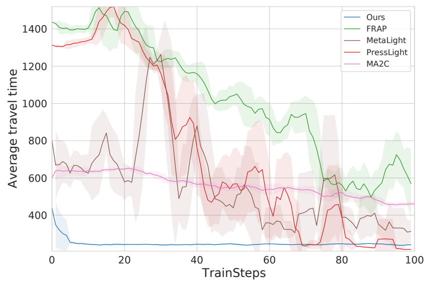

(e) MAN

Fig. 4. Performances of different methods on three real-world data. Columns (a-d) represent different traffic patterns in Hangzhou city.

TABLE I
DIFFERENT CONFIGURATIONS OF CITIES TRAFFIC DATA.

Datasets	Cities	Num of Intersection	Traffic Flow (Veh/s)
Train	HZ	16	0.83
Test	HZ	16	1.82
	HZ	16	1.94
	JN	16	1.21
	MAN	48	0.78

measure to evaluate the performance of the TSC method in transportation. Specifically, ATT is defined as the average time difference between when all vehicles enter the network and when they leave the network.

B. Baselines

We present a comparative evaluation of our proposed method against the following baseline methods including both conventional transportation and RL methods. These can be mainly divided into three categories: 1) **Traditional TSC methods:** SOTL [12] is a conventional method that utilizes current traffic and is controlled with demand-responsive rules that compare the current phase with current traffic. MaxPressure [13] aims to balance the queue length between neighboring intersections by minimizing the pressure of the phases and is currently the state-of-the-art conventional method for network-level TSC. 2) **Online RL methods:** PressLight [14] uses the max-pressure theory and designs the pressure as the reward of the agents, which has good performance in multi-intersection TSC. MA2C [15] uses multi-agent A2C to control multi-intersections cooperatively,

TABLE II

Overall performance. Average travel time is reported in the unit of seconds. (Best values are in bold).

Method	Train Config	Test Config			
	HZ1	HZ2	HZ3	JN	MAN
SOTL	753.3	645.6	747.3	537.5	2711.4
MaxPressure	374.7	420.4	418.6	348.7	305.9
PressLight	341.2	445.2	406.5	320.8	**237.9**
MA2C	340.9	491.1	420.2	397.6	457.5
FRAP	354.3	456.6	414.7	326.4	619.2
MetaLight	354.1	536.2	447.0	328.1	322.7
Offline RL	353.4	526.8	435.7	524.3	934.6
Ours	**339.5**	**413.5**	**400.9**	**307.7**	241.6

including information on neighborhood and spatial discount to stabilize the training process. FRAP [11] framework aims to enhance data efficiency and robustness in traffic signal control by designing a network structure that captures the phase competition relation between different traffic movements. MetaLight [3] is a meta-RL-based method that learns to adapt to different traffic scenarios by utilizing a meta-learner to update the policy parameters. 3) **Offline RL method:** [5] is an offline RL method that learns the FRAP-based policy from datasets without online refinement.

C. Results

1) Overall Performance: We conduct experiments to compare our method with existing meta-RL methods and an online RL method that learns from scratch. To this end, we utilize datasets collected from Hangzhou to train and fine-tune our method in either the original city or the new one. As shown in Fig. 4(a-c), we conduct experiments in the same city but with different traffic patterns. Experimental results show that our method and Meta-Light outperform the algorithm trained from scratch, both in the initial performance, time to converge, and the final learned performance. These findings provide the effectiveness of pre-training and meta-learning. However, we observe that Meta-Light struggles to achieve satisfactory performance, particularly in challenging maps when tested on different cities. This limitation is attributed to its disregard for the diversity of Meta Learner and inconsistency during migration to new environments. Conversely, our method takes these factors into consideration, thereby maintaining optimal performance in terms of initial performance, time to converge, and final learned performance when tested in new cities. We will illustrate the statement later in the ablation experiments.

We show the evaluation performance of our method and all baselines as well as conventional TSC and RL-based TSC in Table II. The proposed method outperforms most methods in the five different scenarios, leading to the least travel time of all vehicles. It is worth mentioning that Offline RL can achieve good results on the train configuration, but it does not perform well on the test configuration, which also illustrates the importance of online refinement.

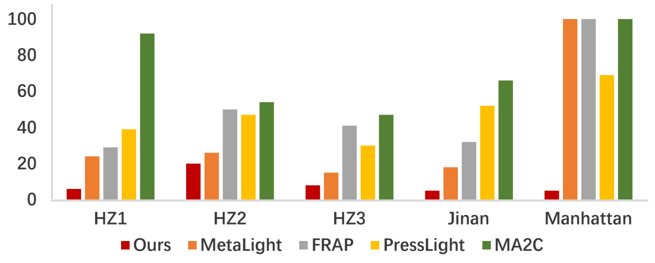


Fig. 5. Number of Episodes to Converge.

TABLE III

The initial performance without finetune.

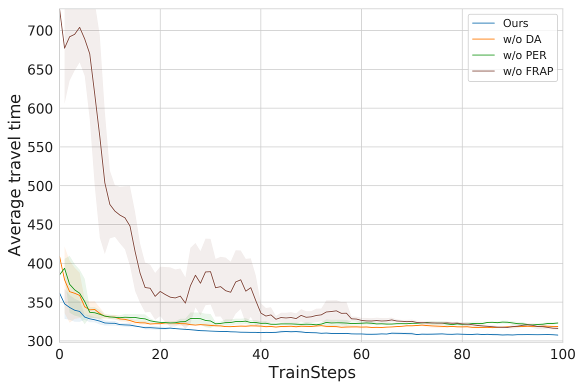
Maps	HZ1	HZ2	HZ3	Jinan	Manhattan	Mean
w/ DA	368.7	**492.7**	**424.1**	**339.5**	**743.5**	**473.7**
w/o DA	**353.4**	526.8	435.7	524.3	934.6	554.9

2) Sample Efficiency: As previously mentioned, the primary challenge in implementing RL-based TSC approaches in real-world scenarios is the impracticality of trial-and-error for environmental interaction. Hence, there is a need for a method that exhibits both high sample efficiency and good initial performance. Specifically, the ideal approach would require minimal interactions to achieve satisfactory performance levels. To assess the efficacy of our pre-training and fine-tuning approach in enhancing sample efficiency, we evaluate our framework against other RL-based Traffic Signal Control (TSC) methods that lack a pre-training phase. As shown in Figure 5, we present the number of interactions with the environment required by each method to attain the convergence performance. Our pre-trained model exhibits considerably lower interaction requirements than the baselines, thereby validating the efficacy of our approach in improving sample efficiency.

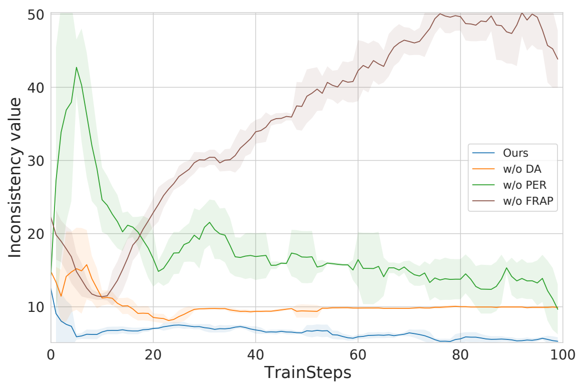
D. Ablation Study

Here, we investigate how each component's effectiveness affect our method's performance. As shown in Fig. 6(a), the performance without consistency PER (w/o CPER), FRAP-based model or DA (w/o DA) is worse than our method, which confirms the usefulness of our method. We quantitatively demonstrate the effectiveness of CPER in our method. As shown in Fig. 6(b), we report the trend of the inconsistency value during the training process. We found that CPER can help the process of online refinement quickly converge, as they emphasize the transition with high inconsistency value and correct the dynamic gap between the pre-trained models and the online scenarios.

As mentioned in Section IV-A, DA serves as a simple technique that can help to reduce overfitting and improve the generalization performance of the pre-trained model. To investigate the effectiveness of DA, we report the initial performance of the pre-trained model without the online refinement. As shown in Table III, when we tested the pre-trained model in different scenarios, i.e., HZ2, HZ3, Jinan and Manhattan, we found that the initial performance of the



(a) Components analysis



(b) Inconsistency value

Fig. 6. Ablation Study. (a) Performance on Jinan Roadnet with and without Consistency Prioritized Experience Replay (CPER), FRAP-based model or Data Augmentation (DA). (b) The trend of the inconsistency value during the training process.

method with DA is higher than that of the method without DA. We think the phenomenon comes from DA providing a more diverse dataset and improving the generalization ability of the pre-trained model to various traffic scenarios.

VI. CONCLUSIONS

In this paper, we proposed a novel offline-to-online RL framework for TSC that provides a potential direction for real-world deployment of RL-based TSC. By utilizing pre-collected datasets and the FRAP network structure, we are able to pre-train a generalized model that can be adapted to various traffic scenarios. The online refinement process with a discrepancy measure and priority sample further improves the agent's performance in new scenarios with minimal online interactions. Our experimental results demonstrate the effectiveness and efficiency of our proposed approach in comparison to several state-of-the-art methods, especially in terms of effectively reducing the training time and computational resources needed for learning in new traffic scenarios.

REFERENCES

[1] H. Wei, G. Zheng, H. Yao, and Z. Li, "Intellilight: A reinforcement learning approach for intelligent traffic light control," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2018, pp. 2496–2505.

[2] 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," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 04, 2020, pp. 3414–3421.

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

[4] J. Ma and F. Wu, "Feudal multi-agent reinforcement learning with adaptive network partition for traffic signal control," *arXiv preprint arXiv:2205.13836*, 2022.

[5] S. Fujimoto, E. Conti, M. Ghavamzadeh, and J. Pineau, "Benchmarking batch deep reinforcement learning algorithms," *arXiv preprint arXiv:1910.01708*, 2019.

[6] S. Fujimoto and S. S. Gu, "A minimalist approach to offline reinforcement learning," *Advances in Neural Information Processing Systems*, vol. 34, 2021.

[7] A. Kumar, J. Fu, M. Soh, G. Tucker, and S. Levine, "Stabilizing off-policy q-learning via bootstrapping error reduction," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[8] Y. Wu, G. Tucker, and O. Nachum, "Behavior regularized offline reinforcement learning," *arXiv preprint arXiv:1911.11361*, 2019.

[9] J. Ma and F. Wu, "Learning to coordinate from offline datasets with uncoordinated behavior policies," in *Proceedings of the 2023 International Conference on Autonomous Agents and Multiagent Systems*, 2023, pp. 1258–1266.

[10] H. Sun and F. Wu, "Less is more: Refining datasets for offline reinforcement learning with reward machines," in *Proceedings of the 2023 International Conference on Autonomous Agents and Multiagent Systems*, 2023, pp. 1239–1247.

[11] 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*, 2019, pp. 1963–1972.

[12] S.-B. Cools, C. Gershenson, and B. D'Hooghe, "Self-organizing traffic lights: A realistic simulation," in *Advances in applied self-organizing systems*. Springer, 2013, pp. 45–55.

[13] P. Varaiya, "The max-pressure controller for arbitrary networks of signalized intersections," in *Advances in Dynamic Network Modeling in Complex Transportation Systems*. Springer, 2013, pp. 27–66.

[14] H. Wei, C. Chen, G. Zheng, K. Wu, V. Gayah, K. Xu, and Z. Li, "Presslight: Learning max pressure control to coordinate traffic signals in arterial network," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2019, pp. 1290–1298.

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

[16] J. Ma and F. Wu, "Feudal multi-agent deep reinforcement learning for traffic signal control," in *Proceedings of the 19th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, Auckland, New Zealand, May 2020, pp. 816–824.

[17] S. Levine, A. Kumar, G. Tucker, and J. Fu, "Offline reinforcement learning: Tutorial, review, and perspectives on open problems," *arXiv preprint arXiv:2005.01643*, 2020.

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

[19] E. D. Cubuk, B. Zoph, J. Shlens, and Q. V. Le, "Randaugment: Practical automated data augmentation with a reduced search space," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, 2020, pp. 702–703.

[20] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *International conference on machine learning*. PMLR, 2020, pp. 1597–1607.

[21] M. Laskin, K. Lee, A. Stooke, L. Pinto, P. Abbeel, and A. Srinivas, "Reinforcement learning with augmented data," *Advances in neural information processing systems*, vol. 33, pp. 19 884–19 895, 2020.

[22] S. Sinha, A. Mandlekar, and A. Garg, "S4rl: Surprisingly simple self-supervision for offline reinforcement learning in robotics," in *Conference on Robot Learning*. PMLR, 2022, pp. 907–917.

[23] T. Schaul, J. Quan, I. Antonoglou, and D. Silver, "Prioritized experience replay," *arXiv preprint arXiv:1511.05952*, 2015.