CrossLight: Offline-to-Online Reinforcement Learning for Cross-City Traffic Signal Control

Qian Sun
The Division of Emerging Interdisciplinary Areas, The Hong Kong University of Science and Technology
Hong Kong SAR, China
qsunal@connect.ust.hk

Rui Zha
School of Computer Science, University of Science and Technology of China
Hefei, China
crui0210@gmail.com

Le Zhang*
Jingbo Zhou
Baidu Research, Baidu Inc.
Beijing, China
zhangle0202@gmail.com
zhoujingbo@baidu.com

Yu Mei
Department of Intelligent Transportation System, Baidu Inc.
Beijing, China
whqyqy@hotmail.com

Zhiling Li
Department of Intelligent Driving Group Business Management, Baidu Inc.
Beijing, China
lizhiling01@baidu.com

Hui Xiong*
Thrust of Artificial Intelligence, The Hong Kong University of Science and Technology (Guangzhou), China
Department of Computer Science and Engineering, The Hong Kong University of Science and Technology
Hong Kong SAR, China
xionghui@ust.hk

ABSTRACT

The recent advancements in Traffic Signal Control (TSC) have highlighted the potential of Reinforcement Learning (RL) as a promising solution to alleviate traffic congestion. Current research in this area primarily concentrates on either online or offline learning strategies, aiming to create optimized policies for specific cities. Nevertheless, the transferability of these policies to new cities is impeded by constraints such as the limited availability of high-quality data and the expensive and risky exploration process. To this end, in this paper, we present an innovative cross-city Traffic Signal Control (TSC) paradigm called CrossLight. Our approach involves meta training using offline data from source cities and adaptively fine-tuning in the target city. This novel methodology aims to address the challenges of transferring TSC policies across different cities effectively. In our proposed approach, we start by acquiring meta-decision pattern knowledge through trajectory dynamics reconstruction via pre-training in source cities. To address disparities in road network topologies between cities, we dynamically construct city topological structures based on the extracted meta-knowledge during the offline meta-training phase. These structures are then used to distill pattern-structure aware representations of decision trajectories from the source cities. To identify effective initial parameters for the learnable components, we employ the Model-Agnostic Meta-Learning (MAML) framework, a popular meta-learning approach. During adaptive fine-tuning in the target city, we introduce a replay buffer that is iteratively updated using online interactions with a rank and filter mechanism. This mechanism, along with a carefully designed exploration strategy, ensures a balance between exploitation and exploration, thereby fostering both the diversity and quality of the trajectories for fine-tuning. Finally, extensive experiments across four cities validate that CrossLight achieves comparable performance in new cities with minimal fine-tuning iterations, surpassing both existing online and offline methods. This success underscores that our CrossLight framework emerges as a groundbreaking and potent paradigm, offering a feasible and effective solution to the intelligent transportation community.

CCS CONCEPTS

• **Computing methodologies** → **Control methods**; • **Applied computing** → **Transportation**.

KEYWORDS

Reinforcement Learning; Traffic Signal Control

ACM Reference Format:
Qian Sun, Rui Zha, Le Zhang, Jingbo Zhou, Yu Mei, Zhiling Li, and Hui Xiong. 2024. CrossLight: Offline-to-Online Reinforcement Learning for Cross-City Traffic Signal Control. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '24), August 25–29, 2024, Barcelona, Spain.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3637528.3671927

1 INTRODUCTION

The increasing urbanization and the growing number of vehicles on the roadways have presented significant challenges such as traffic congestion in many urban centers worldwide. This issue

*Corresponding authors.

KDD '24, August 25–29, 2024, Barcelona, Spain

ACM ISBN 979-8-4007-0490-1/24/08
https://doi.org/10.1145/3637528.3671927

not only prolongs travel times for commuters but also negatively impacts the environment and the economy [1, 21, 29]. As a result, intelligent Traffic Signal Control (TSC) emerges as a pivotal solution to this issue, serving as an indispensable mechanism to mitigate congestion in urban cities [20, 26, 34].

Prior studies have attempted Reinforcement Learning (RL) based methods for TSC [26, 29, 32]. These approaches, taking the traffic system as the environment and the traffic signals as agents, and leveraging the structural dependencies among traffic signals, have consistently demonstrated superior performance over traditional rule-based approaches. However, due to the nature of learning from trail-and-errors, these RL methods require an extremely large number of online exploration trails to gather sufficient samples to learn a near-optimal policy. Moreover, such trial-and-errors are highly risky as they could lead to severe congestion or accidents as a sacrifice for model training, limiting their practical applications in real-world scenarios due to safety concerns.

Taking these issues into account, offline RL has recently been considered as an alternative solution for TSC [15, 34, 38]. These algorithms, which utilize either batch RL [12] or sequential decision modeling [31], are designed to learn effective policies from fixed datasets. These methods have demonstrated impressive performance by mitigating the risks of online interactions. However, a major limitation of offline RL is its dependency on previously collected data from expert models, including records of states, actions, and rewards. When considering the deployment of TSC systems in new cities, the assumption of having such comprehensive trajectory datasets readily available is often impractical.

The aforementioned limitations of the state-of-the-art online and offline RL models for TSC hinder their adaption to new city scenarios due to the necessity of costly explorations and the deficiency of high-quality offline dataset. Hence, in this paper, we propose a novel and feasible solution that fully leverages existing offline datasets collected from readily deployed source cities to train a generalized TSC policy and transfer it to the target city, so-called *Cross-City Transfer*. However, direct cross-city transfer possesses several challenges: (1) The trajectory dynamics of the decision sequences differ significantly between the training and adaptation scenarios, so the policy learnt from the source cities is not generalizable to the target city. The distribution shift between cities can render the strategy developed for source cities ineffective when applied to a target city. (2) The spatial dependencies among traffic signals vary across cities, leading to structural deviations during policy transfer. Typical spatial message passing schemes applied in RL-based TSC rely on the specific network structure of a city, so they struggle to adapt when applied to a new city with an unseen topology graph. (3) Furthermore, fine-tuning the learned policy in the target city becomes challenging when there is a lack of readily available trajectory data. Gathering such data through online interactions can be costly and risky, so it is crucial to determine a strategy that utilizes a minimal amount of interactions data for effective few-shot fine-tuning.

To address the aforementioned challenges for cross-city TSC transfer, we design an offline-to-online meta-learning framework that trains on the source cities offline data to obtain a transferable meta knowledge and adaptively fine-tunes on the target city in a few-shot manner through online interactions with the environment, as illustrated in Figure 1. Specifically, to derive a generalized decision-making pattern from the dynamic trajectories of the source cities that could aid in extracting decision patterns in the target city, we first pre-train a meta decision pattern extraction encoder on the source cities. This process is aimed at distilling meta decision pattern knowledge, facilitating the adaptation and application of learned insights to the target city's context. During the meta-training stage, based on the node-level decision pattern knowledge retrieved with the trajectories data, an adjacency graph is constructed to tackle the city structure discrepancy among different source cities. With the reconstructed structural relationships, the pattern-structure aware representations are obtained, based on which the TSC decisions are auto-regressively predicted through a causal transformer decoder. During the adaptation stage, we develop a novel online few-shot fine-tuning scheme to imitate real-world deployment of TSC policies in a new city. Specifically, besides an exploration strategy during online interactions, an adaptive replay buffer with a rank and filter scheme is designed to facilitate efficient few-shot fine-tuning.

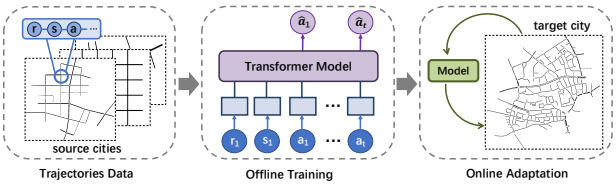

Figure 1: Offline-to-Online Cross-City Transfer.

The major contributions of this paper are summarized as follows:

- To the best of our knowledge, we are among the first to study the cross-city transfer problem in the traffic signal control field, which can bridge the gap between RL-based TSC approaches and their practical implementation in real-world urban scenarios.
- We develop a novel framework that combines offline meta-training with online adaptation to facilitate the transfer of traffic signal control policies across cities, considering both patterns in decision trajectory dynamics and structural dependencies among signals, along with strategies for efficient action exploration and trajectory filtering to ensure data quality for effective fine-tuning.
- Extensive experiments on four city-level datasets demonstrate the effectiveness of our model in terms of transferring TSC policies from source cities to a target city, with a minimal amount of online data required for fine-tuning. Our framework serves as a feasible solution for real-world deployment of TSC in new urban scenarios.

2 RELATED WORKS

2.1 Traffic Signal Control

Traffic signal control focuses on the strategic manipulation of traffic signal phase plans to alleviate congestion. Traditional TSC methods adjust the phases based on traffic information collected in the previous timestep [7, 22, 27]. For instance, MaxPressure [22] is a rule-based scheme that dictates the subsequent phase based on the

"pressure" from the preceding timestep, which is defined as the difference in vehicular flows between incoming and outgoing lanes. Recognizing the dynamic nature of traffic patterns, many studies have formulated the TSC task as a RL problem and such models have achieved superior performance compared to traditional methods [1, 5, 30, 44]. For instance, multi-agent RL models with Graph Neural Networks are proposed to allow collaboration among neighboring intersections [26, 29]. Although they have demonstrated superior performance on multiple datasets, real-world deployment of such models is infeasible due to the uncertainties introduced by the trial-and-error process inherent in online RL training.

Driven by the limitations of online-RL methods, recent works have shifted towards offline approaches [15, 34, 38]. These methods derive decision-making strategies from pre-collected demonstration trajectories without access to an online environment. For instance, [15] and [38] introduce batch RL based methods that approximate a policy from interaction datasets consisting of tuples of state, action, and reward trajectories collected by training expert policies. [34] employs a sequential decision modeling method for TSC, leveraging the Decision Transformer architecture [6]. Results have shown that such offline models can obtain convincing performance compared to online models given sufficient high-quality data. However, the key issue of these offline RL methods is the necessity of training an online model in the target city to gather the demonstration dataset, which is infeasible during real-world deployment. Moreover, the performance of these methods is not only limited by the optimality of the behavior policy employed to collect the data, but also sensitive to shifts between the training and evaluation data distributions [14].

2.2 Reinforcement Learning from Offline to Online

To address the aforementioned data distributional shifts, the RL community has started to investigate methods that employ further online interactions to finetune the pretrained offline RL models, so-called Offline-to-Online(O2O) RL [11, 13, 19, 43]. Such methods focus on finding out the way to best utilize the offline dataset to minimize the number of online interactions for learning the optimal policy [2, 24, 42]. For example, [11] studies an implicit Q-learning algorithm for offline RL that demonstrates strong online finetuning performance, [19] studies an offline pre-trained TD3-BC model with online finetuning using TD3 with an online replay buffer, [43] extends the sequential modeling based offline RL model by incorporating an online replay buffer to further finetune the offline Decision Transformer model and has demonstrated promising performance improvement. Nonetheless, these methods primarily investigate scenarios in which the trajectories encountered during offline training originate from an environment identical to that experienced during online fine-tuning. The problem of cross-scenario transfer remains an under-explored topic in the O2O literature.

2.3 Knowledge Transfer Across Cities

Recently, several works have started to investigate cross-scenario transfer, i.e., knowledge transfer to tackle transfer learning from data-rich scenarios to data-scarce scenarios, especially in the urban computing domain [10, 17, 18, 25, 35]. For example, [10] introduces the region correlation between the source cities and the target city to accomplish the cross-city transfer for traffic forecasting. [18] proposes a graph few-shot learning framework that generates the parameters of spatio-temporal forecasting networks based on the learnt meta-knowledge. [17] designs a cross-city few-shot traffic forecasting framework by introducing a traffic pattern bank to match similar traffic patterns across different cities. Nonetheless, none of the existing studies on knowledge transfer across cities have addressed the transfer of RL policies for traffic signal control.

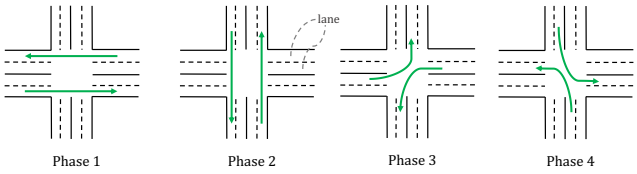

Figure 2: Example of a four-phase intersection.

3 PRELIMINARY

The goal of our task is to improve the travel efficiency in the target city with minimal online interactions by transferring the well-trained offline TSC policy from pre-collected demonstration dataset of existing source cities. In this section, we first define the TSC task and the RL formulation for TSC, and then we introduce the formulation of offline RL in terms of sequence modeling, lastly we formulate our cross-city transfer problem.

3.1 RL for Traffic Signal Control

For TSC, we study the change in phase plan of traffic lights. As shown in Figure 2, a green phase refers to a particular interval during which traffic movements in specific directions are permitted. Decision making in TSC either focuses on fixing the duration of each phase and controlling the order of green phases, or maintaining a predefined order of green phases and determining the time to switch to the next phase, aiming at reducing congestion at the intersections. In RL, the environment can be modeled as a Markov Decision Process (MDP) described by the tuple $(\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma)$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ represents the action space, $\mathcal{P}$ stands for the transition probability matrix, and $\mathcal{R}$ and γ are the reward function and reward discount factor [21, 40]. For RL-based TSC, the state space $\mathcal{S}$ is formulated by a combination of traffic features on different incoming and outgoing lanes of the intersections. Actions $\mathcal{A}$ dictate either the current phase duration given fixed phase plans, or the subsequent green phase index, provided a minimum green phase duration [26]. Reward metrics $\mathcal{R}$ are congestion indicators such as total queue length in incoming directions, average travel time, average waiting time, total throughput, etc [1, 30].

3.2 Offline RL via Sequence Modeling

Offline decision-making learns exclusively from a fixed dataset, bypassing online interactions. The dataset consists of trajectories composed of states s, actions a, and rewards r formatted as follows:

$$\tau = (s_1, a_1, r_1, s_2, a_2, r_2, \ldots, s_T, a_T, r_T), \tag{1}$$

where T is the predefined trajectory length, ranging from 1 to the maximum episode length. In RL-based decision making, the objective is to formulate a decision policy that maximizes the expected return $\mathbb{E}[\sum_{t=1}^{T} r_t]$ in an MDP. Recently, a few works have approached

the offline decision making problem from the supervised learning perspective, utilizing sequential models as opposed to policy gradients or Q-function learning. For instance, Decision Transformer [6] introduces the concept of target return, i.e., $\tilde{r}_t = \sum_{t'=t}^{K} r_{t'}$, serving to evaluate the cumulative reward from a specific timestep t to the trajectory end. Accordingly, the MDP tokens can be transformed to the input sequence below:

$$\tau = (\tilde{r}_1, s_1, a_1, \tilde{r}_2, s_2, a_2, ...\tilde{r}_K, s_K, a_K)\,. \tag{2}$$

On such basis, the objective then shifts to predicting the next action a_t based on the historical MDP tokens $\tau_{1:t}$. Along this line, sequence modeling techniques such as Transformers [23] can be utilized to achieve effective offline decision-making.

3.3 Cross-City TSC Policy Transfer

A city traffic signal structure graph can be denoted as $G = (V, \mathcal{E}, A, X)$. V is the set of traffic lights, $\mathcal{E}$ represents the set of edges, each denoted by $e_{ij} = (v_i, v_j)$, $A \in \mathbb{R}^{N\times N}$ is the adjacency matrix of G, where $N = |V|$, and $a_{ij} = 1$ indicates existence of a connection between traffic signal v_i and v_j. We denote T as the total number of time steps, $X \in \mathbb{R}^{N\times 3K}$ represents the node feature that contains the trajectories data formatted in Equation 2.

Definition 1 (**Cross-city Few-Shot Policy Transfer**). *Given M source cities, i.e., $G_{source} = \{G^1_{source}, \ldots, G^M_{source}\}$ with exclusive offline trajectories data and a target city G_{target} with no existing offline trajectories but limited access to interactions with the environment, the goal of cross-city few-shot policy transfer is to learn a model based on the available trajectories from G_{source} and a minimal amount of online few-shot trajectories from G_{target} to generate TSC actions that maximize the travel efficiency in G_{target}.*

4 METHODOLOGY

In this section, we introduce the technical details of our CrossLight framework, which is illustated in Figure 4. Specifically, we first describe the RL setup of our TSC task, and then introduce our design of the trajectory dynamics reconstruction module in which a decision pattern extractor is pre-trained to obtain node-level meta decision knowledge. We then describe the city structure construction module that constructs the adjacency relationships from the node-level meta decision knowledge, which is then utilized to obtain the pattern-structure aware representations. Then we introduce the causal transformer decoder for auto-regressive action prediction based on the learnt representations. Lastly we introduce the specific offline-to-online meta learning architecture design in our framework that tackles few-shot online adaptation in the target city with an exploration and trajectory rank and filter strategy.

4.1 Reinforcement Learning Setup

We first formulate the TSC task as a Markov Decision Process as introduced in Section 3.1. Particularly, the state space contains the number of approaching vehicles, queue length of stopped vehicles, average speed of approaching vehicles, and pressure (difference between total number of waiting vehicles in the upstream incoming lanes and that in the downstream outgoing lanes [28]) in 12 total traffic flow directions i.e., *N-S*, *N-W*, *N-E*, *S-N*, *S-W*, *S-E*, *W-N*, *W-S*, *W-E*, *E-N*, *E-W*, *E-S*, as well as the current phase encoding (one-hot vector indicating the current green phase in the phase cycle). The action space is the set of non-conflicting directions to be assigned green, in total there are six possible phase actions, assuming the minimum green phase duration t_G is fixed. For the reward function, negative average waiting time at each intersection is selected. Formally, the offline TSC problem for a city c with a total of N^c intersections can be formulated as follows: given the offline dataset collected from each intersection, the trajectory sequences are formulated by global constructs represented by $\tau^c_R \in \mathbb{R}^{K\times N^c\times 1}$, $\tau^c_S \in \mathbb{R}^{K\times N^c\times F_S}$, and $\tau^c_A \in \mathbb{R}^{K\times N^c\times F_A}$, where K denotes the length of trajectory sequence, F_S denotes the feature space dimension of states, and F_A represents the dimension of action space, given the discrete actions are pre-processed into one-hot vectors.

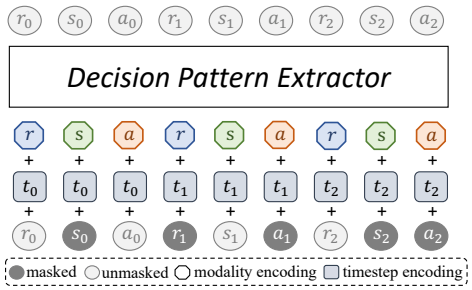

Figure 3: Trajectory Dynamics Reconstruction.

4.2 Trajectory Dynamics Reconstruction

In order to extract a generalized decision pattern from the source cities that could be transferred to the target city, we design a meta decision pattern extractor that is pre-trained on the trajectories data from the source cities with a masking pattern specifically designed for the hetero-modal decision trajectories. Inspired by recent progress in self-supervised pre-training with masked autoencoding [3, 16, 33, 37], we adopt a random auto-regressive masking pattern to align with offline decision sequence modeling. Particularly, we force at least one token in the masked sequence to be auto-regressive, which means the token should be predicted based only on previous tokens, with all future tokens masked. Particularly, for each input trajectory τ we first randomly choose a modality $m_0 \in \{S, A, R\}$, then a random index $k_{m_0} \in [1, K]$ is selected for modality m_0. For tokens of modality m_0, those starting from timestep k_{m_0} are masked, i.e., $\tau^{k_{m_0}:K}_{m_0}$ are masked. Meanwhile, for the other two modalities m_1 and m_2, trajectories τ_{m_1} and τ_{m_2} are randomly masked, following a predefined masking ratio. An example masking pattern is shown in Figure 3, where $m_0 = A$ and $k_{m_0} = 1$. With such masking pattern, the auto-encoder can learn the underlying temporal dependencies across the tokens and perform inference in an auto-regressive manner.

For the decision pattern extractor, we employ the commonly adopted encoder-decoder architecture where both components are bidirectional transformers [23]. Notably, given the diverse dimensionalities of the input tokens $\tau^c_S, \tau^c_A, \tau^c_R$, we first attempt to map these tokens into a unified representation space by employing fully connected layers $f_s(\cdot)$, $f_a(\cdot)$, and $f_r(\cdot)$ respectively, resulting in

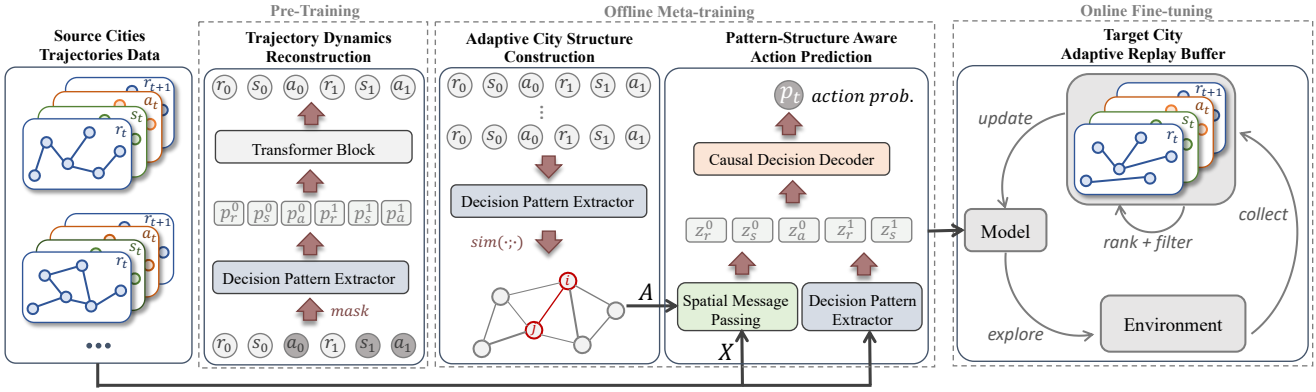

Figure 4: An overview of CrossLight, which includes three main phases: (1) In the pre-training stage, masked trajectory sequences are reconstructed with the Decision Pattern Extractor. (2) Within offline meta-training, the city structure is reconstructed and then fed to the Pattern-Structure Aware Action Prediction module to auto-regressively predict actions. (3) During online fine-tuning, an Adaptive Replay Buffer is iteratively updated during online interactions with explorations.

representations $\{H_S^c, H_A^c, H_R^c\} \in \mathbb{R}^{K\times N^c\times D}$. To enable the transformers to differentiate between tokens with different modalities at different timesteps in the sequence, we integrate a learnable mode-specific encoding as well as a sinusoidal timestep encoding on top of the input token encodings. The decoder is then trained to reconstruct the original sequence, including the unmasked tokens, using the MSE loss. The training objective for the self-supervised pretraining for the trajectory dynamics reconstruction is thus:

$$\max_{\theta} \mathbb{E} \sum_{t=1}^{T} \sum_{m=1}^{3} \log P_\theta(z_m^t | \tau_{\text{masked}}) \tag{3}$$

where τ_{masked} is the masked trajectory sequence. In such way, the decision pattern extractor is able to acquire effective representations of the input trajectories which represents the generalized decision pattern knowledge from the source cities.

4.3 City Structure Construction

Given the input trajectories $\tau_R^c, \tau_S^c, \tau_A^c$ from the source city c, we query the pre-trained encoder in the decision pattern extractor to obtain the pattern-aware representations, i.e., $\{P_R^c, P_S^c, P_A^c\} \in \mathbb{R}^{K\times N^c\times D}$ where D is the hidden dimension of the output encoding from the encoder. We then define the node-level pattern knowledge by concatenating the representations from different modalities:

$$H_{MK}^c = P_R^c \parallel P_S^c \parallel P_A^c, \tag{4}$$

where $\parallel$ represents the concatenation operation.

Besides decision patterns across timesteps, structural inter-signal communications has also been shown a crucial factor in TSC in many existing works [26, 29]. In order to express the structural information of different city graphs and reduce the structural deviations, we propose to reconstruct the adjacency relationship from the node-level pattern knowledge H_{MK}. Specifically, we use H_{MK} to predict the probability of edge existence between each pair of nodes $(n_i, n_j) \in \mathbb{G}^c$. Formally, the constructed adjacency matrix can be obtained by:

$$\hat{A}^c = \text{sigmoid}\left([H^{MK}]^T W \cdot H^{MK}\right), \tag{5}$$

where $(\cdot)^T$ is the transpose operation, and $W \in \mathbb{R}^{D\times D}$ is the learnable weight matrix. Given the reconstructed graph structure relationship driven from the hidden pattern knowledge from the source cities, we further proceed to obtain structure-aware representations leveraging spatial message passing among the intersections.

4.4 Structure Aware Representations

A conventional approach for spatial message passing is to employ Graph Neural Networks (GNNs), such as GCN [39], on the predefined road network to capture the inherent spatial patterns and correlations. However, given that GNNs are primarily adept at modeling local topological information, directly relying on adjacency relationships between nodes might not adequately capture the dynamically changing spatial correlations among traffic signals. Therefore, we adopt a transformer-like architecture to handle the representations, without introducing additional inductive bias.

Specifically, we first introduce a learnable spatial position encoding tailored to each type of tokens. The encodings can be represented as $\{E_R^C, E_S^C, E_A^C\} \in \mathbb{R}^{N\times N}$, which is initialized with the road network adjacency matrix $\hat{A}^c$, where N is the maximum number of nodes in the source cities. Subsequently, the encodings are concatenated with the hidden token representations $\{H_S^c, H_A^c, H_R^c\}$. Formally, the encoded representations can be derived by:

$$\tilde{H}_{R,S,A}^c = f\left(H_{R,S,A}^c \parallel E_{R,S,A}^c\right), \tag{6}$$

where $f(\cdot)$ stands for the linear mapping, and $\parallel$ denotes the concatenation operation. Along this line, we further leverage a spatially-oriented multi-head attention with residual connections to capture the latent spatial dependencies among different traffic signals:

$$\tilde{Z}_{R,S,A}^c = \text{MHA}\left(\tilde{H}_{R,S,A}^c\right) + \tilde{H}_{R,S,A}^c, \tag{7}$$

where $\text{MHA}(\cdot)$ represents the well-known Multi-Head Attention operation [23]. Parallelly, a multi-layer GCN with residual connections is applied on the hidden representations $H_{R,S,A}^c$ to obtain the

static spatially aggregated representations. The message propagation for layer $\ell + 1$ within GCN is provided as follows:

$$H^{(\ell+1)} = \sigma\left(D^{c-\frac{1}{2}}\hat{A}^c D^{c-\frac{1}{2}} H^{(\ell)} W^{(\ell)} + H^{(\ell)}\right), \tag{8}$$

where D^c is the diagonal degree matrix of A^c, $W^{(\ell)}$ is the learnable weight matrix, σ is the activation function.

Denoting the spatially aggregated representation outputs from GCN as $\hat{Z}^c_{R,S,A}$, the final structure-aware token representations can be obtained by applying a gate mechanism with no bias, as follows:

$$\begin{aligned} g &= \sigma\left(\tilde{f}\left(\tilde{Z}^c_{R,S,A}\right) + \hat{f}\left(\hat{Z}^c_{R,S,A}\right)\right), \\ Z^c_{R,S,A} &= g \odot \tilde{Z}^c_{R,S,A} + (1-g) \odot \hat{Z}^c_{R,S,A}, \end{aligned} \tag{9}$$

where σ is the sigmoid activation function, $\hat{f}$ and $\tilde{f}$ represent linear mappings, and $\odot$ denotes element-wise multiplication. On such basis, we can adaptively uncover the intricate interconnections inherent among the traffic signals in the hidden embeddings.

4.5 Pattern-Structure Aware Action Prediction

Then, the meta decision patterns and the structure-aware representations Z^c_R, Z^c_S, Z^c_A are combined to obtain the pattern-structure auxiliary representations for decision prediction, as shown by:

$$\mathcal{Z}^c_{R,S,A} = H^c_{R,S,A} \parallel Z^c_{R,S,A}. \tag{10}$$

Then, the pattern-structure aware representations $\mathcal{Z}^c_R$, $\mathcal{Z}^c_S$, and $\mathcal{Z}^c_A$ are transformed into the following sequence:

$$\tau_z = \left(z^1_r, z^1_s, z^1_a, z^2_r, ..., z^t_r, z^t_s, z^t_a\right), z_{r,s,a} \in \mathcal{Z}^c_{R,S,A}. \tag{11}$$

Given the success of Decision Transformer [6] in offline decision sequence modeling, we adopt a similar transformer architecture for causal action prediction. Based on the embedding sequence re-ordered in the way that is readily available for return-guided decision modeling, we apply a Transformer decoder architecture that auto-regressively predicts the action probabilities p^c at timestep t based on previous tokens from timestep 1 up to t:

$$p^c_t = \text{TransformerDecoder}(\tau^{1:t}_z). \tag{12}$$

Accordingly, we employ the cross-entropy loss as the optimization objective since the phase actions are discrete in our task, the loss is defined as follows:

$$\mathcal{L} = -\sum_{t=1}^{T}\sum_{i=1}^{P} p^c_{t,i} \log\left(a^c_{t,i}\right), \tag{13}$$

where P stands for the total number of phase actions.

4.6 Offline Meta-training and Online Finetuning

To handle few-shot adaptation in the target city, we propose a meta-learning framework, originating from Model-Agnostic Meta-Learning(MAML) [8], which contains a meta-training process in the source cities and an adaptation process in the target city. However, different from traditional meta-learning scenarios where both the training and adaptation stage require offline data, our task is special since only the meta-training stage is purely offline, while no offline trajectories are readily available in the adaptation stage for finetuning. Tailored to this situation, we first define each source task $\mathcal{T}_i \in \mathcal{T}$ that includes D_S support set τ_{spt} and D_Q query set τ_{qry}, where $\tau_{spt} \bigcap \tau_{qry} = \emptyset$. With regard to training task $\mathcal{T}_i$, task-specific model parameters $\theta_{\mathcal{T}_i}$ is updated by gradient descents for several steps using the support set, as shown below:

$$\theta'_{\mathcal{T}_i} = \theta_{\mathcal{T}_i} - \alpha \nabla_{\theta_{\mathcal{T}_i}} L_{\tau_{spt}}\left(f_{\theta_{\mathcal{T}_i}}\right). \tag{14}$$

Then the model is evaluated on the query set, where the accumulated gradients across the query sets over all tasks are used to train the general model parameters θ :

$$\theta \leftarrow \theta - \beta \nabla_\theta \sum_{\mathcal{T}_i} L_{\tau_{qry}}\left(f_{\theta'_{\mathcal{T}_i}}\right). \tag{15}$$

Further, in the adaptation stage, to guarantee finetuning on the target city where offline trajectories do not readily exist, we design an iteratively updated replay buffer $\mathcal{B}$ that collects trajectories from the target city while interacting with the environment. Since the decision model $\mathcal{F}_\theta$ is well-trained on the source cities trajectories which are different from the target city in terms of data distribution, directly applying $\mathcal{F}_\theta$ to generate actions might lead to sub-optimal trajectories that harm the model fine-tuning performance. Hence, we propose an exploration strategy on the deterministic $\mathcal{F}_\theta$ model, particularly Boltzmann exploration [4]. Specifically, during online finetuning and data collection, instead of deterministically selecting the action with the maximum action probability, we sample the action from a categorical distribution based on the action probability output from $\mathcal{F}_\theta$, following:

$$a_t \sim \text{Categorical}\left(\mathcal{F}_\theta\left(\tau^{1:t}_{target}\right)\right), \tag{16}$$

where τ_{target} are the iterative trajectory segments from the target city. This design strikes a balance between exploration and exploitation by making higher-valued actions more likely to be chosen while still allowing lower-valued actions to be explored. Moreover, to guarantee the quality of the target trajectories data, we apply a trajectory filtering scheme that ranks the trajectories in $\mathcal{B}$ by mean rewards of the sequences, and keep the highest $k\%$ trajectories in $\mathcal{B}$. Furthermore, based on the filtered trajectories, we update the decision model $\mathcal{F}_\theta$ by fine-tuning using samples from $\mathcal{B}$, following:

$$\theta = \theta - \lambda \nabla_\theta L_{\mathcal{B}}\left(f_\theta\right). \tag{17}$$

The entire offline-to-online training-finetuning procedure is shown in Algorithm 1 in details.

5 EXPERIMENTS

5.1 Simulation and Datasets

We utilize SUMO[1] (Simulation of Urban MObility), a widely adopted microscopic multi-modal traffic simulator, to simulate traffic dynamics including phase changes of traffic signals and movements of vehicles. We conduct experiments on four city-level data consisting of road networks and traffic flow configurations, namely *Grid-4x4* [1], *Jinan* [20], *Hangzhou* [20], and *Baoding* [21]. The data descriptions are provided in Table 1. To gather the demonstration datasets, we train the representitive CoLight model [29] for 100

[1]Website: https://www.eclipse.org/sumo/

Algorithm 1: Offline Meta-training and Online Fine-tuning

Input: decision model $\mathcal{F}_\theta$, pretrained decison pattern extractor P_ϕ, source city data D_{source}, source tasks set $\mathcal{T}$, replay buffer $\mathcal{B}$, environment **E**.
Output: finetuned decision model parameters θ.

```
// Offline meta training on source city tasks
```
for *epoch in range(1, meta_epochs)* **do**
  for $\mathcal{T}_i \in \mathcal{T}$ **do**
    $\theta_{\mathcal{T}_i} \leftarrow \theta$.
    $\tau_{spt} \leftarrow SampleSupport(\mathcal{T}_i, D_{source})$.
    $\tau_{qry} \leftarrow SampleQuery(\mathcal{T}_i, D_{source})$.
    Compute action prediction loss on the support set, $L_{\tau_{spt}}(f_{\theta_{\mathcal{T}_i}})$ based on Equation (13).
    Update the task model parameters with gradient descent following Equation (14).
    Evaluate the updated $\theta'_{\mathcal{T}_i}$ on the query set via Equation (13).
  Update θ based on Equation (15).

```
// Online few-shot finetuning in the target city
```
$env \leftarrow SetupEnv(\mathbf{E})$
for *epoch in range(1, finetune_epochs)* **do**
  Rollout trajectory τ_{target} from *env* using the decision model $\mathcal{F}_\theta$ following Equation (16).
  Append the trajectories to the replay buffer: $\mathcal{B} \leftarrow \mathcal{B} \cup \{\tau_{target}\}$.
  Filter $\mathcal{B}$ by the top k-percent rewards: $\mathcal{B} \leftarrow RankandFilter(\mathcal{B}, k)$.
  Update $\mathcal{F}_\theta$ using $\mathcal{B}$ via Equation (17).

epochs on the first hour(3600s) traffic flow for all cities. We set $t_G = 10s$, hence obtaining a total of 360 updates of states, actions, and rewards within each epoch. We select trajectories from the final 50 epochs after carefully analyzing the model convergence curve for all cities to ensure the quality of the demonstration datasets while conserving trajectory dynamics.

5.2 Benchmark Methods

The compared baselines include the following: **FixedTime**, where the traffic signal phase plan switches in the pre-defined order; **MaxPressure** [22], which adaptively selects the next phase based on current intersection pressure; **Colight** [29], which uses a multi-agent DQN model for neighbor-aware TSC; **STMARL** [26], which leverages DQN for TSC considering the spatio-temporal dependencies among intersections; **MetaLight** [36], which is a value-based meta-reinforcement learning model for TSC; **DataLight** [38], which is a conservative Q-learning model for offline TSC that trains the policy from pre-collected trajectory datasets; **TransformerLight** [34]: which is a shared Decision Transformer model for all traffic light agents in the city network; **DTLight** [9]: which is a decentralized Decision Transformer model with online fine-tuning on the specific city networks; **SO2** [41]: which is a smoothed O2O method that improves the Q-value estimation by perturbing the target action and improving the frequency of Q-value updates.

Table 1: Dataset Descriptions

Dataset	Grid4x4	Hangzhou	Jinan	Baoding
Synthetic/Real	Synthetic	Real	Real	Real
Number of Intersections	16	16	12	21
Total Vehicle Flow	1,473	6,984	6,295	1,466
Expert Converge Epoch	42	44	52	48

Since none of the existing TSC models study cross-city transfer, we train and evaluate the above-mentioned baselines on the target city. For our model implementation, we regard the city being evaluated as the target city, while the other cities as source cities.

5.3 Experimental Settings

Evaluation Metrics Commonly used evaluation metrics for TSC include vehicles' average waiting time, average travel time, total throughput, average queue lengths, etc [1, 20, 30]. In this paper, we select three metrics to evaluate the performance, including *average delay* (defined as $t_{real} - t_{expected}$, where t_{real} is the actual trip duration of a vehicle and $t_{expected}$ is the ideal trip time if the vehicle travels at its maximum speed in the traffic network without any traffic restrictions, e.g., traffic signals), *average waiting time*, and *average travel time*.

Implementation Details In our experiments, we set the sequence length to 4, hidden dimension to 256, mask ratio to 0.75, and batch size is selected to be 256, meta training steps is 5, online finetuning epochs is 30, and replay buffer filtering percentage is selected to be 50%, the learning rate is selected as $1e^{-4}$. For performance evaluation, we evaluate the models on the target city for 10 epochs and report the average values.

5.4 Performance Evaluation

We report the comparative analysis between our CrossLight framework and the baseline models including heuristic rule-based methods, online RL, and offline RL approaches in Table 2. The results clearly demonstrate that CrossLight outperforms competing methods on the investigated datasets in terms of nearly all of the three evaluation metrics measuring total traffic efficiency. Firstly, the best-performing rule-based baseline *MaxPressure* is outperformed by 51.02%, 44.11%, 28.53%, and 82.80% in terms of average delay, on the four cities respectively. Both serving as feasible solutions for real-world TSC deployment, our cross-city transfer framework outperforms the best-performing rule-based baseline, showcasing the promise of our solution. Moreover, our framework also outperforms most online RL-based baselines. For example, comparing with the structure-aware models *CoLight* and *STMARL*, our model outperforms *CoLight* by 16.61% and 43.44% on *Grid4x4* and *Baoding* in terms of average waiting time, and 10.15% and 61.98% improvements can be seen compared to *STMARL* in terms of average travel time. Furthermore, our model outperforms the meta-learning based baseline *MetaLight* by 9.03%, and 26.08% on *Grid4x4* and *Baoding* in terms of average travel time. Such results demonstrate that our cross-city model is comparable to existing online models, indicating the significance of our offline training and online finetuning design. On the other hand, for offline RL baselines, our model achieves 1.71%, and 1.03% improvements on the *Hangzhou* dataset compared

Table 2: The overall performance of different models on four city-level datasets, where AD denotes Average Delay, AWT denotes Average Waiting Time, and ATT denotes Average Travel Time. All metrics are in seconds.

City	Grid4x4			Hangzhou			Jinan			Baoding		
Metrics	AD	AWT	ATT	AD	AWT	ATT	AD	AWT	ATT	AD	AWT	ATT
FixedTime	76.51	36.90	213.04	61.87	18.21	433.27	96.02	48.92	334.68	1027.33	1010.33	1148.12
MaxPressure	48.22	22.92	159.56	47.59	11.98	349.84	70.51	34.61	297.20	854.87	772.83	1004.79
CoLight	40.22	9.63	141.79	42.36	3.72	335.40	64.41	21.84	279.16	237.94	217.87	423.57
STMARL	38.28	10.05	143.28	40.91	4.12	332.48	63.15	22.01	279.54	232.10	209.54	378.46
MetaLight	42.68	12.98	141.51	45.79	6.76	342.78	64.37	21.98	280.41	168.54	149.76	194.67
DataLight	25.51	9.94	130.02	28.98	3.16	335.71	63.57	24.78	**274.04**	175.46	146.38	225.00
TransformerLight	30.16	10.18	141.75	29.62	3.18	333.42	62.65	27.50	289.73	181.88	144.23	229.26
DTLight	26.67	8.12	137.61	39.23	8.40	340.51	57.20	34.06	278.95	164.62	157.28	162.43
SO2	28.70	8.93	135.08	39.84	6.51	333.29	59.48	29.75	281.02	156.37	149.93	157.89
CrossLight	**23.62**	**8.03**	**128.73**	**26.60**	**3.12**	**329.98**	**50.39**	**20.78**	279.52	**147.06**	**123.22**	**143.90**

Table 3: Ablation performance of Average Delay.

City	Grid4x4	Hangzhou	Jinan	Baoding
w/o pattern extraction	25.85	28.07	54.16	151.40
w/o structure construction	24.78	28.91	53.76	152.01
w/o spatial message passing	28.92	31.74	58.92	169.70
w/o online finetuning	46.97	52.28	81.73	560.91
CrossLight	**23.62**	**26.60**	**50.39**	**147.06**

to the batch RL based baseline *DataLight* and the sequential modeling driven model *TransformerLight* in terms of average travel time. Moreover, compared to the offline-to-online baseline *DTLight*, our model improves by 62.85% and 21.66% on *Hangzhou* and *Baoding* in terms of average waiting time. These satisfactory results demonstrate the effectiveness of cross-city meta knowledge learning from the source cities and transfer to the target city. Such cross-city transfer could potentially achieve similar or higher performance than end-to-end training in the target city, which is often infeasible. This demonstrates the effectiveness of meta-training on the source cities that provides a good initialization for fine-tuning on the target city that guarantees performance. Such results are sufficient to draw the conclusion that our cross-city transfer framework offers a better solution that is both feasible in real-world settings and well-performing in terms of travel efficiency maximization in the target city.

In terms of ablation studies, we investigate the model performance with removal of different modules in our framework, and we provide the results in Table 3. The results demonstrate the effectiveness of the sub-modules in our framework, specifically, the most important component in our framework is the online finetuning with real-time adaptations in the target city, resulting in 49.71% and 73.78% improvements on *Grid4x4* and *Baoding*. Additionally, spatial message passing is also an essential aspect, signified by 16.19% and 14.48% improvements on *Hangzhou* and *Jinan*.

5.5 Model Analysis

In this section, we evaluate CrossLight from various perspectives, addressing the following research questions one by one:

RQ1: *How does the scale and diversity of the source cities' data affect the cross-city transfer performance in the target city?*

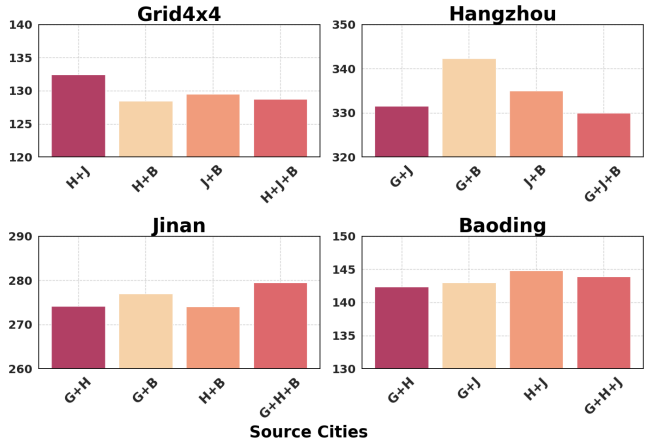

Figure 5: Model performance of Average Travel Time with different source city trajectories composition.

In order to address this question, we conduct experiments with different combinations of source cities to evaluate the impact of different source cities on the effectiveness of cross-city transfer. The results are visualized in Figure 5, in which the x-axis shows different combinations of the source cities, where *G* stands for *Grid4x4*, *H* represents *Hangzhou*, *J* stands for *Jinan*, and *B* is abbreviated for *Baoding*. From Figure 5, for *Grid4x4*, *Hangzhou+Baoding* provides the best source cities combination, followed by the three-cities combination, this is potentially due to the similarity between the traffic flow distributions between *Grid4x4* and *Baoding*, which is also validated by the fact that performance of transferring from *Grid4x4+Hangzhou* results in the best performance when evaluating on *Baoding*. For *Hangzhou*, the superior performance when transferring from *Grid4x4+Jinan* can be explained by the similar structural mappings between *Grid4x4* and *Hangzhou* as well as the similar traffic flow patterns between *Jinan* and *Hangzhou*.

RQ2: *How robust is our framework to parameters selection such as sequence length K?*

In order to answer this research question, we conduct a parameter sensitivity analysis with different selections of sequence length, and provide the results visualization on *Grid4x4* in Figure

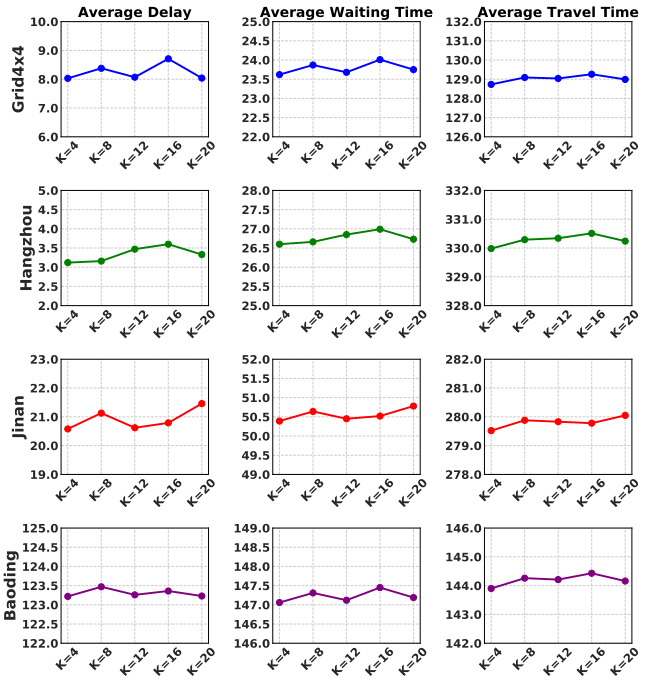

Figure 6: Model performance with different sequence length.

6. From the results we can draw the conclusion that our model is robust against different choices of sequence lengths, and the best-performing option is $K = 4$. Similar conclusion can be drawn from the other datasets.

6 CONCLUSION

In this paper, we proposed CrossLight, a novel offline-to-online reinforcement learning framework for cross-city traffic signal control. Particularly, we designed a meta-learning based framework that meta-trains on the offline trajectories data from the source cities, and fine-tunes on the target city with the online trajectories adaptively gathered through interactions with the environment. Starting with pre-training a decision pattern extractor that is capable of acquiring generalized pattern knowledge, we then meta-trained on the source cities data while constructing the city structure graphs. We then developed an online few-shot adaptation scheme with exploration and trajectories filtering strategies to facilitate fine-tuning in the target city. Experimental results on four city-level datasets have demonstrated the effectiveness of our framework. Overall, our framework represents a significant step forward in bridging the gap between RL-based TSC approaches and the practical implementation of TSC policies in new urban settings.

7 ACKNOWLEDGMENTS

This work was supported by the National Natural Science Foundation of China (Grant No.92370204), National Key R&D Program of China (Grant No.2023YFF0725001), Guangzhou-HKUST(GZ) Joint Funding Program (Grant No.2023A03J0008), Education Bureau of Guangzhou Municipality.

REFERENCES

[1] James Ault and Guni Sharon. 2021. Reinforcement learning benchmarks for traffic signal control. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 1)*.

[2] Philip J. Ball, Laura Smith, Ilya Kostrikov, and Sergey Levine. 2023. Efficient online reinforcement learning with offline data. In *Proceedings of the 40th International Conference on Machine Learning* (Honolulu, Hawaii, USA) *(ICML'23)*. JMLR.org, Article 67, 18 pages.

[3] Micah Carroll, Orr Paradise, Jessy Lin, Raluca Georgescu, Mingfei Sun, David Bignell, Stephanie Milani, Katja Hofmann, Matthew Hausknecht, Anca Dragan, et al. 2022. Uni [mask]: Unified inference in sequential decision problems. *Advances in neural information processing systems* 35 (2022), 35365–35378.

[4] Nicolò Cesa-Bianchi, Claudio Gentile, Gábor Lugosi, and Gergely Neu. 2017. Boltzmann exploration done right. *Advances in neural information processing systems* 30 (2017).

[5] Chacha Chen, Hua Wei, Nan Xu, Guanjie Zheng, Ming Yang, Yuanhao Xiong, Kai Xu, and Zhenhui Li. 2020. Toward a thousand lights: Decentralized deep reinforcement learning for large-scale traffic signal control. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 3414–3421.

[6] Lili Chen, Kevin Lu, Aravind Rajeswaran, Kimin Lee, Aditya Grover, Misha Laskin, Pieter Abbeel, Aravind Srinivas, and Igor Mordatch. 2021. Decision transformer: Reinforcement learning via sequence modeling. *Advances in neural information processing systems* 34 (2021), 15084–15097.

[7] Seung-Bae Cools, Carlos Gershenson, and Bart D'Hooghe. 2013. Self-organizing traffic lights: A realistic simulation. *Advances in applied self-organizing systems* (2013), 45–55.

[8] Chelsea Finn, Pieter Abbeel, and Sergey Levine. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. In *International conference on machine learning*. PMLR, 1126–1135.

[9] Xingshuai Huang, Di Wu, and Benoit Boulet. 2023. Traffic Signal Control Using Lightweight Transformers: An Offline-to-Online RL Approach. *arXiv preprint arXiv:2312.07795* (2023).

[10] Yilun Jin, Kai Chen, and Qiang Yang. 2022. Selective cross-city transfer learning for traffic prediction via source city region re-weighting. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 731–741.

[11] Ilya Kostrikov, Ashvin Nair, and Sergey Levine. 2021. Offline reinforcement learning with implicit q-learning. *arXiv preprint arXiv:2110.06169* (2021).

[12] Sascha Lange, Thomas Gabel, and Martin Riedmiller. 2012. Batch reinforcement learning. In *Reinforcement learning: State-of-the-art*. Springer, 45–73.

[13] Seunghyun Lee, Younggyo Seo, Kimin Lee, Pieter Abbeel, and Jinwoo Shin. 2022. Offline-to-online reinforcement learning via balanced replay and pessimistic q-ensemble. In *Conference on Robot Learning*. PMLR, 1702–1712.

[14] Sergey Levine, Aviral Kumar, George Tucker, and Justin Fu. 2020. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *arXiv preprint arXiv:2005.01643* (2020).

[15] Jianxiong Li, Shichao Lin, Tianyu Shi, Chujie Tian, Yu Mei, Jian Song, Xianyuan Zhan, and Ruimin Li. 2023. A Fully Data-Driven Approach for Realistic Traffic Signal Control Using Offline Reinforcement Learning. *arXiv preprint arXiv:2311.15920* (2023).

[16] Fangchen Liu, Hao Liu, Aditya Grover, and Pieter Abbeel. 2022. Masked autoencoding for scalable and generalizable decision making. *Advances in Neural Information Processing Systems* 35 (2022), 12608–12618.

[17] Zhanyu Liu, Guanjie Zheng, and Yanwei Yu. 2023. Cross-city Few-Shot Traffic Forecasting via Traffic Pattern Bank. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*. 1451–1460.

[18] Bin Lu, Xiaoying Gan, Weinan Zhang, Huaxiu Yao, Luoyi Fu, and Xinbing Wang. 2022. Spatio-Temporal Graph Few-Shot Learning with Cross-City Knowledge Transfer. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 1162–1172.

[19] Yicheng Luo, Jackie Kay, Edward Grefenstette, and Marc Peter Deisenroth. 2023. Finetuning from Offline Reinforcement Learning: Challenges, Trade-offs and Practical Solutions. *arXiv preprint arXiv:2303.17396* (2023).

[20] Hao Mei, Xiaoliang Lei, Longchao Da, Bin Shi, and Hua Wei. 2022. LibSignal: An Open Library for Traffic Signal Control. *arXiv preprint arXiv:2211.10649* (2022).

[21] Qian Sun, Le Zhang, Huan Yu, Weijia Zhang, Yu Mei, and Hui Xiong. 2023. Hierarchical reinforcement learning for dynamic autonomous vehicle navigation at intelligent intersections. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 4852–4861.

[22] Pravin Varaiya. 2013. Max pressure control of a network of signalized intersections. *Transportation Research Part C: Emerging Technologies* 36 (2013), 177–195.

[23] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems* 30 (2017).

[24] Andrew Wagenmaker and Aldo Pacchiano. 2023. Leveraging offline data in online reinforcement learning. In *Proceedings of the 40th International Conference on Machine Learning* (Honolulu, Hawaii, USA) *(ICML'23)*. JMLR.org, Article 1470, 39 pages.

[25] Leye Wang, Xu Geng, Xiaojuan Ma, Feng Liu, and Qiang Yang. 2019. Cross-city Transfer Learning for Deep Spatio-temporal Prediction. In *IJCAI International Joint Conference on Artificial Intelligence*. 1893.

[26] Yanan Wang, Tong Xu, Xin Niu, Chang Tan, Enhong Chen, and Hui Xiong. 2020. STMARL: A spatio-temporal multi-agent reinforcement learning approach for cooperative traffic light control. *IEEE Transactions on Mobile Computing* 21, 6 (2020), 2228–2242.

[27] Fo Vo Webster. 1958. *Traffic signal settings*. Technical Report.

[28] Hua Wei, Chacha Chen, Guanjie Zheng, Kan Wu, Vikash Gayah, Kai Xu, and Zhenhui Li. 2019. Presslight: Learning max pressure control to coordinate traffic signals in arterial network. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*. 1290–1298.

[29] Hua Wei, Nan Xu, Huichu Zhang, Guanjie Zheng, Xinshi Zang, Chacha Chen, Weinan Zhang, Yanmin Zhu, Kai Xu, and Zhenhui Li. 2019. Colight: Learning network-level cooperation for traffic signal control. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*. 1913–1922.

[30] Hua Wei, Guanjie Zheng, Vikash Gayah, and Zhenhui Li. 2021. Recent advances in reinforcement learning for traffic signal control: A survey of models and evaluation. *ACM SIGKDD Explorations Newsletter* 22, 2 (2021), 12–18.

[31] Muning Wen, Runji Lin, Hanjing Wang, Yaodong Yang, Ying Wen, Luo Mai, Jun Wang, Haifeng Zhang, and Weinan Zhang. 2023. Large sequence models for sequential decision-making: a survey. *Frontiers of Computer Science* 17, 6 (2023), 176349.

[32] Libing Wu, Min Wang, Dan Wu, and Jia Wu. 2021. DynSTGAT: Dynamic spatial-temporal graph attention network for traffic signal control. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*. 2150–2159.

[33] Philipp Wu, Arjun Majumdar, Kevin Stone, Yixin Lin, Igor Mordatch, Pieter Abbeel, and Aravind Rajeswaran. 2023. Masked trajectory models for prediction, representation, and control. *arXiv preprint arXiv:2305.02968* (2023).

[34] Qiang Wu, Mingyuan Li, Jun Shen, Linyuan Lü, Bo Du, and Ke Zhang. 2023. TransformerLight: A Novel Sequence Modeling Based Traffic Signaling Mechanism via Gated Transformer. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining* (Long Beach, CA, USA).

[35] Huaxiu Yao, Yiding Liu, Ying Wei, Xianfeng Tang, and Zhenhui Li. 2019. Learning from multiple cities: A meta-learning approach for spatial-temporal prediction. In *The world wide web conference*. 2181–2191.

[36] Xinshi Zang, Huaxiu Yao, Guanjie Zheng, Nan Xu, Kai Xu, and Zhenhui Li. 2020. Metalight: Value-based meta-reinforcement learning for traffic signal control. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 1153–1160.

[37] Rui Zha, Ying Sun, Chuan Qin, Le Zhang, Tong Xu, Hengshu Zhu, and Enhong Chen. 2024. Towards Unified Representation Learning for Career Mobility Analysis with Trajectory Hypergraph. *ACM Transactions on Information Systems* 42, 4 (2024), 1–28.

[38] Liang Zhang and Jianming Deng. 2023. Data Might be Enough: Bridge Real-World Traffic Signal Control Using Offline Reinforcement Learning. *arXiv preprint arXiv:2303.10828* (2023).

[39] Si Zhang, Hanghang Tong, Jiejun Xu, and Ross Maciejewski. 2019. Graph convolutional networks: a comprehensive review. *Computational Social Networks* 6, 1 (2019), 1–23.

[40] Weijia Zhang, Hao Liu, Jindong Han, Yong Ge, and Hui Xiong. 2022. Multi-agent graph convolutional reinforcement learning for dynamic electric vehicle charging pricing. In *Proceedings of the 28th ACM SIGKDD conference on knowledge discovery and data mining*. 2471–2481.

[41] Yinmin Zhang, Jie Liu, Chuming Li, Yazhe Niu, Yaodong Yang, Yu Liu, and Wanli Ouyang. 2024. A Perspective of Q-value Estimation on Offline-to-Online Reinforcement Learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 38. 16908–16916.

[42] Han Zheng, Xufang Luo, Pengfei Wei, Xuan Song, Dongsheng Li, and Jing Jiang. 2023. Adaptive policy learning for offline-to-online reinforcement learning. *arXiv preprint arXiv:2303.07693* (2023).

[43] Qinqing Zheng, Amy Zhang, and Aditya Grover. 2022. Online decision transformer. In *international conference on machine learning*. PMLR, 27042–27059.

[44] Liwen Zhu, Peixi Peng, Zongqing Lu, and Yonghong Tian. 2023. Metavim: Meta variationally intrinsic motivated reinforcement learning for decentralized traffic signal control. *IEEE Transactions on Knowledge and Data Engineering* (2023).