X-Light: Cross-City Traffic Signal Control Using Transformer on Transformer as Meta Multi-Agent Reinforcement Learner

Haoyuan Jiang[1] , **Ziyue Li**[2,†] , **Hua Wei**[3] , **Xuantang Xiong**[4] , **Jingqing Ruan**[4] , **Jiaming Lu**[5] , **Hangyu Mao**[6] and **Rui Zhao**[6]
[1]Baidu Inc., China
[2]University of Cologne, Germany
[3]Arizona State University, U.S.A
[4]Institute of Automation, Chinese Academy of Sciences, China
[5]Fudan University, China
[6]SenseTime Research, China
jianghaoyuan@zju.edu.cn, zlibn@wiso.uni-koeln.de.

Abstract

The effectiveness of traffic light control has been significantly improved by current reinforcement learning-based approaches via better cooperation among multiple traffic lights. However, a persisting issue remains: how to obtain a multi-agent traffic signal control algorithm with remarkable transferability across diverse cities? In this paper, we propose a **T**ransformer **on T**ransformer (TonT) model for *cross*-city meta multi-agent traffic signal control, named as X-Light: We input the full Markov Decision Process trajectories, and the Lower Transformer aggregates the states, actions, rewards among the target intersection and its neighbors *within a city*, and the Upper Transformer learns the general decision trajectories *across different cities*. This dual-level approach bolsters the model's robust generalization and transferability. Notably, when directly transferring to unseen scenarios, ours surpasses all baseline methods with **+7.91%** on average, and even **+16.3%** in some cases, yielding the best results.

1 Introduction

An effective traffic signal control (TSC) system is the key to alleviating traffic congestion. In recent years, Reinforcement Learning (RL) has been widely used in the field of TSC. It can interact with the environment, explore, and exploit, which helps agents discover better policies without artificial priors and assumptions. Numerous studies [Wei *et al.*, 2019a; Zheng *et al.*, 2019; Chen *et al.*, 2020; Wei *et al.*, 2019b; Chu *et al.*, 2019; Lu *et al.*, 2023; Du *et al.*, 2024] have demonstrated its noteworthy enhancements over conventional rule-based methods [Hunt *et al.*, 1982; Lowrie, 1990; Roess *et al.*, 2004; Smith *et al.*, 2013].

However, most of the existing methods are *scenario-specific*, meaning that the training and testing should be in the same scenario (A scenario is a simulation environment, e.g., a virtual city, with a set of intersections). When deploying on a new scenario, rebuilding the environment and re-training are needed, which involves significant costs. As a result, to the best of our knowledge, all the cities still use rule-based methods such as SCATS or SCOOT for a very fundamental reason: they can be easily reused in a new district/region/city.

This raised a critical problem that **how to orchestrate the multiple intersections for various scenarios/cities with strong generalizability.**

There are some solutions for single-agent settings: MetaLight [Zang *et al.*, 2020] and GESA [Jiang *et al.*, 2024] used one single agent to control all the intersections in a scenario and achieve transferability via gradient-based meta RL and multi-city co-training, respectively. Their drawbacks are obvious, i.e., neglecting the cooperation among the multiple intersections. Yet, learning a general Multi-Agent RL model for various scenarios is non-trivial, given various scenarios could have various road networks and traffic dynamics, rendering various environment states for each agent and collaboration patterns for multiple agents. To the best of our knowledge, only a few existing works target the same challenge: MetaVIM [Zhu *et al.*, 2023] and MetaGAT [Lou *et al.*, 2022]. They both utilized the Markov Decision Processes (MDPs) trajectories to help the agents learn and distinguish scenario context. However, they still display limitations, such as an unstable training process and large performance drops when encountering quite dissimilar scenarios (details in Sec. 2).

To enhance cooperation and generalizability, we will incorporate the full MDP trajectories, including the observations, rewards, and actions (o, a, r) of both the target intersection and its neighbors, into the method. Given the sequence nature of the trajectories from various MDPs, two natural questions are: **Q1: Can Transformer utilize the** o, a, r **sequences of multiple intersections for better collaboration? Q2: Furthermore, can we Transformer learn the high-level cross-scenario MDPs dynamics for better transferability.** This forms our solution as in Figure 1: a Trans-

This paper's extended version is at *arxiv.org/abs/2404.12090*, with code at *github.com/jianghaoyuan1994/X-Light*

†Corresponding Author

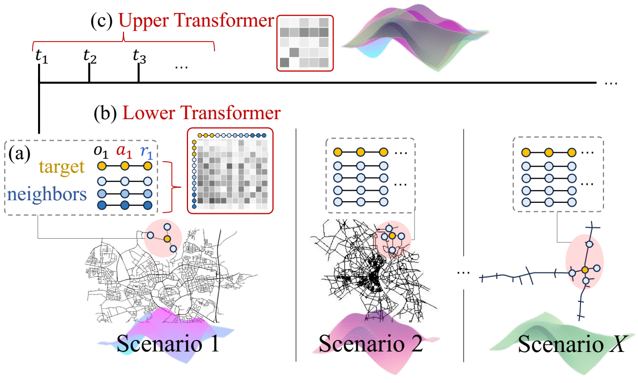


Figure 1: (a) X-Light takes the MDP o, a, r trajectories of the target and its neighbors: (b) the Lower Transformer learns the attention for all the o, a, r-s, so that, e.g., one intersection's o may have high attention with another intersection's a; (c) Upper Transformer learns the attention over the time through all different scenarios.

former on Transformer (TonT) model for Meta Multi-Agent Reinforcement Learners. The **Lower Transformer** extracts single-step trajectory information from the target intersection and its neighbor intersections to encourage cooperation. The **Upper Transformer** learns from the historical multi-scenario multi-modal MDPs distributions and makes actions.

The main contributions are three-fold:

- In the domain of TSC, we propose the first-ever Transformer-on-Transformer (TonT) framework for meta MARL, which solves both multi-intersection collaboration and cross-city transferability/generalizability.
- Specifically, the **Lower Transformer** aggregates the target and its neighbors' fine-grained o, a, r information and achieves better collaboration than other solutions such as the Graph Neural Network (GNN)-based methods [Wei *et al.*, 2019b; Lou *et al.*, 2022], which only aggregates traffic states o; The **Upper Transformer**, together with a *dynamic prediction* pretext task and *multi-scenario co-training scheme*, learns the scenario-agnostic decision process and achieves better cross-city decision. Additionally, a *residual link* is added before inputting to actor-critic for better decision.
- We conduct rigid experiments with various scenarios and zero-shot transfer to each, and our method is constantly the best performer. In non-transfer settings, we also achieve the most top one results.

2 Related Work

2.1 Meta Reinforcement Learning-based TSC

Meta RL-based methods are gaining attention in TSC since they can greatly reduce adaptation costs and have promising performance in new scenarios. GeneraLight [Zhang *et al.*, 2020] and AttendLight [Oroojlooy *et al.*, 2020] try to generalize TSC to different traffic flow patterns: they use GAN [Goodfellow *et al.*, 2014] and two attention models to handle traffic diversity. However, it is not strictly adapting to a new scenario. MetaLight [Zang *et al.*, 2020] can be seen as the pioneering work of training a TSC agent in multiple scenarios: it uses gradient-based meta-reinforcement learning with base function reading each local data and updating local gradients, and meta-learner accumulating the memory and updating the global gradients. GESA [Jiang *et al.*, 2024] instead proposed a general plug-in module (GPI) to make it possible to read various cities' maps without labels and further uses large-scale scenarios to co-train an actor-critic-based agent. But all of them only considered single-agent settings and ignored the cooperation with others. However, to model the cooperative multi-agent under multiple various scenarios is quite technically challenging. Only a few models are proposed, yet their performance is quite unstable and limited.

MetaVIM [Zhu *et al.*, 2023] uses context-based Meta RL to solve the generalization problem and uses intrinsic rewards to cooperate, but it is only trained in one scenario and the state does not contain neighbor information, leading to challenges in performance and adaptability when encountering an environment that is different from the training scenario. MetaGAT [Lou *et al.*, 2022] extends MetaLight with a Graph Attention Network (GAT) to aggregate the neighbors' observations into the target's and Gated Recurrent unit (GRU) to learn the MDP dynamics. However, it has volatile fluctuations during the training process and suboptimal performance. Potential reasons are that (1) it is trained with one scenario after another, but we mix intersections from various scenarios within one batch; (2) Relying on GAT restricts them to only utilize observation information as node feature, overlooking crucial actions and rewards within the MDP. We instead use Lower Transformer to fully construct the relation among o, a, r from multiple intersections; (3) A hybrid model of combining GAT and GRU may be harder to train, and we are an elegant, Transformer-only model.

2.2 Transformers in RL and Other Fields

Offline RL unfolds the MDP process as a sequence of s_t, a_t, r_t, s_{t+1} etc. Thus, works such as Decision Transformer (DT) [Chen *et al.*, 2021], PDiT [Mao *et al.*, 2023], and Trajectory Transformer (TT) [Janner *et al.*, 2021] use Transformer as the backbone, treating decision-making as a next-token prediction, i.e., to predict the next action a_{t+1}. TransformerLight [Wu *et al.*, 2023] got inspired by DT and trained a TSC agent with offline-collected data. However, training with offline-collected data means a heavy workload to prepare high-quality data, and it cannot interact with the environment in real-time, let alone transfer to a new scenario. To increase transferability, PromptDT [Xu *et al.*, 2022] proposes to add few-shot task-specific sequences as the prompt for quick adaptation for new tasks. However, offline data is still needed.

Transformer as Meta RL: TrMRL [Melo, 2022] is the pioneering work that proves Transformer can function as a meta-agent. Similar to the memory reinstatement mechanism, this agent establishes connections between the immediate working memories to construct an episodic memory across the transformer layers iteratively. Our work is the first one that applies a Transformer-based meta agent into the TSC domain; we further extend the original single-agent setting to multi-agent, with several technical challenges overcome by our final TonT solution: the Lower Transformer guides cooperation with neighbors utilizing o, a, r information, and the upper Transformer learns meta-features across tasks, showcasing remarkable generalization capabilities.

3 Methodology

This section delineates our proposed method, X-Light, a general cross-city multi-agent traffic signal control method. We begin by providing an overview of our method, followed by the introduction of the TonT Encoder module, which contains the Lower Transformer and the Upper Transformer. Finally, we will give detailed settings for training. The preliminary information is presented in the Appx. B.

3.1 Overview

As shown in Fig. 2, our method is trained using intersections across various scenarios within a batch. The i-th intersection and its neighbors $\mathcal{N}_i$ are selected, and their MDP trajectories (o, a, r) from time frame $[t-K+1, t]$ are sampled and fed into TonT Encoder module. It is worth noting that to allow the model to handle various intersections, we use the GPI module proposed in [Jiang *et al.*, 2024], which maps various intersections' structure into a unified one, followed by an MLP to obtain $o_t^{\{i, \mathcal{N}_i\}}$. More details are in the Appx. D.

As shown in Fig. 2, the TonT Encoder employs two types of transformers: the Lower Transformer and the Upper Transformer. The primary role of the Lower Transformer is to integrate the target and its neighbors' MPD information at time step t. It enhances agent collaboration compared with GNN-based collaboration by 6%-13%. The Upper Transformer utilizes historical trajectory information as context to infer the current task, thereby achieving improved transferability.

The output of the TonT Encoder is then utilized by both the Actor and Critic to output the policy π for executing the action and to estimate the state value for training.

Similar to [Wei *et al.*, 2021], we choose five features from the observations as states: queue length, current phase, occupancy, flow, and the number of stopping cars. The action is to choose the eight pre-defined phases for the next time interval, as shown in Fig 2. At each time step t, the agent i can choose to execute action a_t^i from available action set $\mathcal{A}^i$ in the next Δt seconds. In our experiments, we set Δt as 15 seconds. The reward r is defined as the weighted sum of queue length, wait time, delay time, and pressure. More details in Appx. D.

3.2 Lower Transformer

The Lower Transformer enhances collaboration between the target i and its n-nearest neighbors $\mathcal{N}_i, \mathcal{N}_i = \{N_1, \ldots, N_n\}$. Existing multi-agent TSC methods only consider the states of neighboring intersections for cooperation, neglecting the complex and valuable interrelations among their observations, actions, and rewards. Considering the variations in road networks and traffic flow interconnections across diverse scenarios, one intersection's action will affect another's state and also vice versa. Thus, only relying on states is inadequate to simulate the neighbors' impacts on the target accurately. This can further result in instability within the learning process, especially during cross-scenario co-training, as we observed in MetaLight and MetaGAT's training process.

Thus, in the Lower Transformer, we utilize the **full** MDP features of the target and its neighbors to enhance the model's understanding of the cooperation. Since we are online RL, at time t, we can only observe immediate o_t and the previous step's a_{t-1}, r_{t-1}. Thus, the ith agent's MDP feature at t is:

$$\mathbf{m}_t^i = (\mathbf{o}_t^i, \mathbf{a}_{t-1}^i, \mathbf{r}_{t-1}^i) \tag{1}$$

Since original $\mathbf{o}, \mathbf{a}, \mathbf{r}$ have different dimensions, we first employ three trainable linear projections, i.e., $\mathbf{E}_o, \mathbf{E}_a, \mathbf{E}_r$, to map all of them to the same dimension of d. Then, the full MDP transition with neighbors concatenated is defined as:

$$\begin{aligned} &\mathbf{M}_t^i = [\mathbf{m}_t^i\mathbf{E}; \mathbf{m}_t^{N_1}\mathbf{E}; \ldots, \mathbf{m}_t^{N_n}\mathbf{E}] \in \mathbb{R}^{3(1+n)\times d} \\ &\text{where } \mathbf{m}_t^i\mathbf{E} = (\mathbf{o}_t^i\mathbf{E}_o, \mathbf{a}_{t-1}^i\mathbf{E}_a, \mathbf{r}_{t-1}^i\mathbf{E}_r) \in \mathbb{R}^{3\times d} \end{aligned} \tag{2}$$

If the number of neighbors is smaller than n, we use zero-padding and add a binary indicator embedding to all $\mathbf{o}_t$ to indicate whether this neighbor exists.

Similar to ViT [Dosovitskiy *et al.*, 2020], a learnable `[class]` token is prepended serving as the global image representation, we also prepend a trainable `[decision]` token $\mathbf{q}_{\text{decision}} \in \mathbb{R}^d$, whose state at the output of the Lower Transformer is used as the intersection representation $\mathbf{c}_t$. We also add standard position embedding [Vaswani *et al.*, 2017] to each input token to retain positional information. Thus, the input to the Lower Transformer $\mathbf{z}_{t,i}^{\text{lower}}$ is:

$$\mathbf{z}_{t,i}^{\text{lower}} = [\mathbf{q}_{\text{decision}}; \mathbf{m}_t^i\mathbf{E}; \mathbf{m}_t^{N_1}\mathbf{E}; \ldots, \mathbf{m}_t^{N_n}\mathbf{E}] + \mathbf{E}_{pos}^{\text{lower}} \tag{3}$$

where $\mathbf{z}_{t,i}^{\text{lower}}, \mathbf{E}_{pos}^{\text{lower}} \in \mathbb{R}^{(3(1+n)+1)\times d}$ and the attention in the Lower Transformer is in $\mathbb{R}^{(3(1+n)+1)\times(3(1+n)+1)}$. Then, we feed $\mathbf{z}_{t,i}^{\text{lower}}$ to the Lower transformer with 3 multi-head self-attention layers, and it outputs a capsulized intersection embedding for further decision-making, denoted as $\mathbf{c}_t^i$.

$$\mathbf{c}_t^i = \text{Lower Transformer}(\mathbf{z}_{t,i}^{\text{lower}}) \tag{4}$$

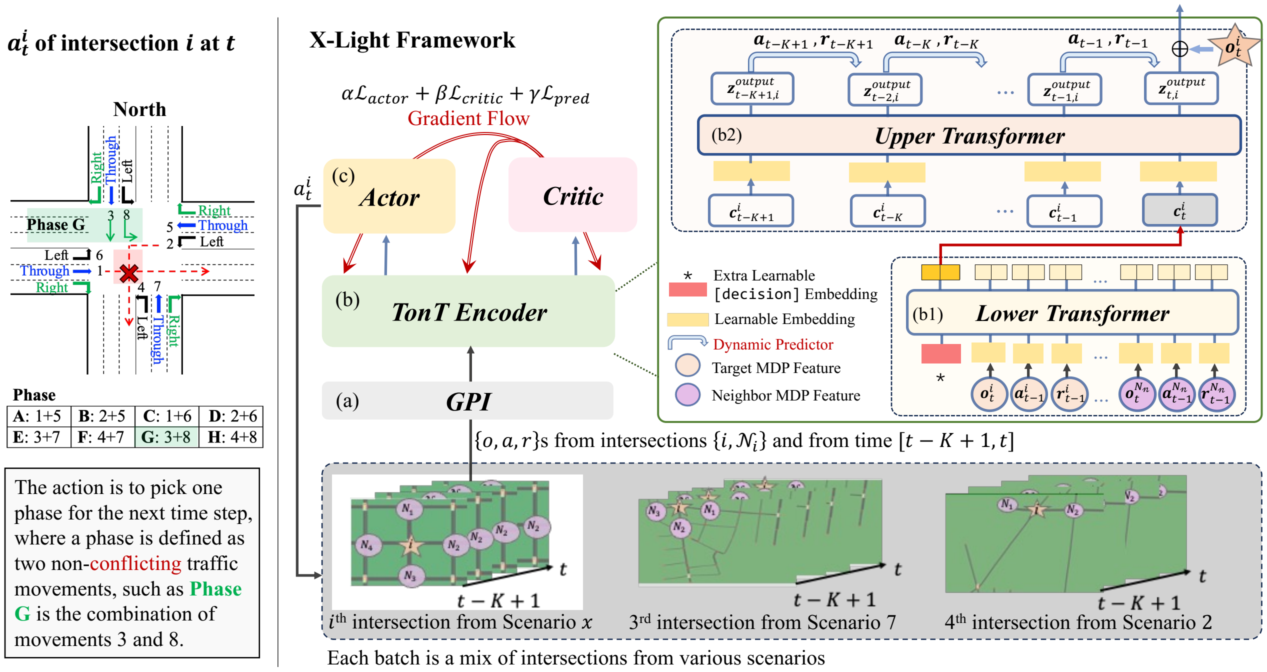
3.3 Upper Transformer

The Upper Transformer ensures that the model exhibits strong generalization in the presence of unseen intersections or scenarios. To achieve this goal, as shown in Fig. 2.(b2), we employ *context-based meta RL* [Duan *et al.*, 2017] and *a dynamic predictor* within the Upper Transformer. $\mathbf{c}_{[t-K+1:t]}^i$ obtained from the Lower Transformer is first projected to dimension-d' through a trainable projection denoted as $\mathbf{E}'$. Then, as the context-based meta RL, the Upper Transformer utilizes these embeddings from the last K time steps. Similarly, along with the positional embedding, the input to the Upper Transformer is $\mathbf{z}_{[t-K+1:t],i}^{\text{upper}} \in \mathbb{R}^{K\times d'}$:

$$\mathbf{z}_{[t-K+1:t],i}^{\text{upper}} = [\mathbf{c}_{t-K+1}^i\mathbf{E}'; \mathbf{c}_{t-K+2}^i\mathbf{E}'; \ldots; \mathbf{c}_t^i\mathbf{E}'] + \mathbf{E}_{pos}^{\text{upper}}, \tag{5}$$

The attention mechanism in the Upper Transformer (consisting of 3 multi-head self-attention layers) operates within the $\mathbb{R}^{K\times K}$ dimension, capturing the environmental dynamics related to the target intersection from historical features.

$$\mathbf{z}_{[t-K+1:t],i}^{\text{output}} = \text{Upper Transformer}(\mathbf{z}_{[t-K+1:t],i}^{\text{upper}}) \tag{6}$$


a_t^i of intersection i at t
North
Right
Through
Left
Phase G
3 8
5 Through
2 Left
Left 6
Through 1
Right
4 7
Left
Through
Right
X-Light Framework
$\alpha\mathcal{L}_{actor} + \beta\mathcal{L}_{critic} + \gamma\mathcal{L}_{pred}$
Gradient Flow
a_t^i
(c) Actor
Critic
(b) TonT Encoder
(a) GPI
$\{o, a, r\}$s from intersections $\{i, \mathcal{N}_i\}$ and from time $[t-K+1, t]$
a_{t-K+1}, r_{t-K+1} a_{t-K}, r_{t-K} a_{t-1}, r_{t-1}
$z^{output}_{t-K+1,i}$ $z^{output}_{t-2,i}$ $z^{output}_{t-1,i}$ $z^{output}_{t,i}$
o_t^i
(b2) Upper Transformer
c^i_{t-K+1} c^i_{t-K} c^i_{t-1} c^i_t
* Extra Learnable [decision] Embedding
Learnable Embedding
Dynamic Predictor
Target MDP Feature
Neighbor MDP Feature
(b1) Lower Transformer
o_t^i a^i_{t-1} r^i_{t-1} $o_t^{N_n}$ $a^{N_n}_{t-1}$ $r^{N_n}_{t-1}$
t
$t-K+1$
i^{th} intersection from Scenario x
3rd intersection from Scenario 7
4th intersection from Scenario 2
Each batch is a mix of intersections from various scenarios

Phase

A: 1+5	B: 2+5	C: 1+6	D: 2+6
E: 3+7	F: 4+7	G: 3+8	H: 4+8

The action is to pick one phase for the next time step, where a phase is defined as two non-conflicting traffic movements, such as Phase G is the combination of movements 3 and 8.

Figure 2: Our method is co-trained with intersections' MDPs from various scenarios: (a) a GPI module unifying all the scenarios, (b) the proposed TonT Encoder, and (c) an actor-critic to make a decision. The TonT Encoder contains (b1) a Lower Transformer aggregating the o, a, and r among the target and its neighbors and (b2) an Upper Transformer learning general decisions from multi-scenario historical MDPs.

Dynamic Predictor: To enhance the agent's comprehension of the current task and the influence between intersections, we introduce a dynamic predictor between the Upper Transformer's each time-step output, enhancing the agent to learn the environment dynamics. The dynamic predictor is a pretext prediction task to conduct autoregression prediction, encouraging the Upper Transformer to capture the cross-scenario dynamics. As shown in Fig. 4, this dynamic predictor can improve performance by **4.4%**. Specifically, we feed the previous time step $\mathbf{z}^{\text{output}}_{t-1,i}$, concatenated with all the actions $\mathbf{a}^{\{i,\mathcal{N}_i\}}_{t-1}$ and rewards $\mathbf{r}^{\{i,\mathcal{N}_i\}}_{t-1}$ from the target and its neighbors, to predict the next $\mathbf{z}^{\text{output}}_{t,i}$:

$$\hat{\mathbf{z}}^{\text{output}}_{t,i} = \text{MLP}([\mathbf{z}^{\text{output}}_{t-1,i}; \mathbf{a}^{\{i,\mathcal{N}_i\}}_{t-1}; \mathbf{r}^{\{i,\mathcal{N}_i\}}_{t-1}]) \tag{7}$$

A prediction loss based on mean squared error (MSE) is:

$$\mathcal{L}_{pred} = \text{MSE}(\hat{\mathbf{z}}^{\text{output}}_{t,i}, \mathbf{z}^{\text{output}}_{t,i}) \tag{8}$$

3.4 Actor-Critic

For the decision policy π, we use the PPO method [Schulman *et al.*, 2017], both Actor and Critic are two-layer MLPs. The actor receives the TonT Encoder's output and makes the action a_i^t for the target intersection i.

Residual Link: However, before inputting into the actor-critic module, to avoid over-abstracting the intersection's embedding through the TonT Encoder, we directly add the observation $\mathbf{o}_t^i$ of the target intersection i into the Upper Transformer's output $\mathbf{z}^{\text{output}}_{t,i}$. This ensures the embedding input into the actor has enough observational information from the target intersection.

$$\mathbf{a}_t^i \sim \pi(\cdot|\mathbf{z}^{\text{output}}_{t,i} + \mathbf{o}_t^i) \tag{9}$$

The $\mathbf{z}^{\text{output}}_{t,i}$ is the embedding after TonT Encoder's feature abstraction, which is particularly essential for scenarios with complex intercorrelations; $\mathbf{o}_t^i$ instead is the direct self-observation, which is proven as rather essential for simple scenarios, where focusing on yourself is enough. As shown in Fig.4, without residual link can lead to a **2%** performance drop. Thus, our design strikes a good balance for scenarios in various complexity. Therefore, together with Eq. (8), the overall optimization objective can be formatted as:

$$\mathcal{L} = \alpha\mathcal{L}_{actor} + \beta\mathcal{L}_{critic} + \gamma\mathcal{L}_{pred}, \tag{10}$$

where α, β, γ are tuning parameters. The Actor loss and Critic loss are the same as PPO.

3.5 Multi-scenario Co-Training

To further increase model generality, we employed the multi-scenario co-train to increase data diversity. Unlike MetaGAT [Lou *et al.*, 2022] or MetaLight [Zang *et al.*, 2020], which utilizes the multi-scenario sequential training, i.e., within a batch, intersections are from the same scenario, and scenarios are read until previous one is all seen, we adopted multi-scenario co-training, i.e., within each batch, intersections are stochastically chosen from various scenarios. Given each intersection from each scenario has a distinct structure, we adopt a general preprocess module, GPI from GESA [Jiang *et al.*, 2024], whose core idea is to map all various-structured intersections into a unified 4-leg one. This encourages the agent to learn generalist knowledge and also enables a more stable training process.

Lastly, the learning algorithm is shown in Algorithm 1.

Algorithm 1 X-Light training process

```
Input: A set of target intersections $\mathcal{I}$ from a set of multi-agent scenarios $\mathcal{X}$; training episodes $E$; the number of neighbor $n$; the Upper Transformer input length $K$;
Initialize: buffer $\mathcal{D}$; parameters $\theta$;
Output: Optimized policy $\pi^{\theta}$
for episode=1, ..., $E$ do
    clean buffer $\mathcal{D} \leftarrow \emptyset$;
    for each scenario $x$ do
        Find nearest $n$ neighbors $\mathcal{N}$ of each intersection;
        for each time step $t$ do
            Get the last $K$ transitions $\{\mathbf{m}^i_{t-k+1}\}^x_{k=1,...,K}$ of each intersection $i$ according to Eq. (1) and add these into $\mathcal{D}$;
            Get action $\mathbf{a}^i_t$ according to Eq. (9) and take joint action $\{\mathbf{a}^1_t, ..., \mathbf{a}^{n_x}_t\}$;
            Get the the next states $\mathbf{o}^i_{t+1}$, and rewards $\mathbf{r}^i_t$;
        end for
    end for
    for each step in training steps do
        sample minibatch data from $\mathcal{D}$;
        Get dynamic predictions $\{\hat{\mathbf{z}}^{\text{output}}_{t-k+1,i}\}_{k=1,...,K-1}$ according to Eq.(7);
        Computer $\mathcal{L}$ according (10) and update parameter $\theta$;
    end for
end for
```

Scenarios	Country	Type	#Total Int.	**Scenarios**	Country	Type	#Total Int.
Grid 4×4	synthetic	region	16	***Cologne8***	Germany	region	8
Avenue 4×4	synthetic	region	16	***Ingolstadt21***	Germany	region	12
Grid 5×5	synthetic	region	25	***Fenglin***	China	corridor	7
				Nanshan	China	region	28

Table 1: Statistics of the scenarios.

4 Experiments

4.1 Datasets

We conducted experiments on the simulation of urban mobility (SUMO) as the simulator. The duration of each episode is 3600 seconds. Seven different scenarios [Ault and Sharon, 2021; Jiang *et al.*, 2024] are employed in our co-training: five of them are reported in results, and another two scenarios (*Fenglin* and *Nanshan*) are only employed to increase the number of meta-training scenarios. To enable multi-scenario co-training, we utilize the GPI module in GESA to significantly reduce the need for manual labeling and unify the observation space and action space of all intersections. The main idea of GPI is to map all variously structured intersections into a unified 4-leg one using the relative angle and masking. For details, please refer to [Jiang *et al.*, 2024]. Table 1 shows the properties of different scenarios. The details about these scenarios are in Appx. E.

4.2 Baselines

To comprehensively verify the effectiveness of our proposed method, we employed four types of methods for a comprehensive comparison:

Conventional methods:

- **Fixed Time Control (FTC)** [Roess *et al.*, 2004] with random offset executes each phase within a loop, utilizing a pre-defined phase duration.
- **MaxPressure** [Kouvelas *et al.*, 2014] is a powerful conventional method that consistently selects the phase with the highest pressure among all phases.

Multi-agent methods:

- **MPLight** [Chen *et al.*, 2020] is based on a phase competition mechanism to select which phase to execute, and utilizes the concept of pressure as state and reward to coordinate multiple intersections.
- **IPPO** [Ault and Sharon, 2021] controls each intersection with an independent PPO agent, and the agents at each intersection have the same model architecture but different model parameters.
- **rMAPPO** [Yu *et al.*, 2022a] is executed with independent PPO agents like IPPO, but it uses the overall information of all intersections to jointly optimize traffic efficiency. Furthermore, we use RNNs to introduce historical information into the agent.
- **CoLight** [Wei *et al.*, 2019b] utilizes the GAT to aggregate information from neighboring intersections, enhancing cooperation between intersections.

Meta-Learning and Single-agent methods:

- **MetaLight** [Zang *et al.*, 2020] using the meta-learning method to train multiple scenarios to increase the generalization of the model.
- **GESA** [Jiang *et al.*, 2024] proposes a unified state and action space, and subsequently employs the GPI module for large-scale multi-scenario collaborative training, which improves performance and generality.

Meta-Learning and Multi-agent methods:

- **MetaGAT** [Lou *et al.*, 2022] combines contextual meta-learning based on GRU to improve generalization and GAT for cooperation. Yet, the GRU is not scenario-agnostic trained, and GAT only uses observations o.
- **X-Light** (ours) and **X-Light$_{\textbf{GNN}}$**, which replaces the Lower Transformer with GNN [Hamilton *et al.*, 2017], and the node feature of each intersection is obtained by concatenating o, a, r into a three times longer embedding, discarding the interrelations among them.

4.3 Evaluation Metrics

As same as most existing works [Lou *et al.*, 2022; Oroojlooy *et al.*, 2020; Chen *et al.*, 2020], we use **average trip time** as a component of evaluation metrics, defined as the average time for each vehicle from entering the scenario to leaving the scenario. However, merely using the average trip time cannot accurately reflect the real traffic situation, e.g., in the cases of severe traffic congestion, new vehicles are prevented from entering the scenario, which leads to a relatively low average trip time though it is a severe traffic condition. Thus, we also add **average delay time** [Ault and Sharon, 2021] for evaluation, defined as the delay caused by signalized intersections and traffic congestion.

Methods	**Avg. Trip Time (seconds)**					**Avg. Delay Time (seconds)**				
	transfer to Grid4×4	*transfer to Grid5×5*	*transfer to Arterial4×4*	*transfer to Ingolstadt21*	*transfer to Cologne8*	*transfer to Grid4×4*	*transfer to Grid5×5*	*transfer to Arterial4×4*	*transfer to Ingolstadt21*	*transfer to Cologne8*
FTC	206.68 ± 0.54	550.38 ± 8.31	828.38 ± 8.17	319.41 ± 24.48	124.4 ± 1.99	94.64 ± 0.43	790.18 ± 7.96	1234.30 ± 6.50	183.70 ± 26.21	62.38 ± 2.95
MP	175.97 ± 0.70	274.15 ± 15.23	686.12 ± 9.57	375.25 ± 2.40	95.96 ± 1.11	64.01 ± 0.71	240.00 ± 18.43	952.53 ± 12.48	275.36 ± 14.38	31.93 ± 1.07
MetaLight	169.21 ± 1.16	244.99 ± 8.18	392.34 ± 4.39	298.67 ± 5.09	92.38 ± 0.94	57.82 ± 0.67	202.32 ± 11.20	850.42 ± 36.35	166.35 ± 5.53	28.37 ± 0.73
GESA	166.23 ± 1.07	284.05 ± 25.36	410.59 ± 2.60	318.30 ± 9.56	88.76 ± 0.46	54.69 ± 0.81	246.96 ± 37.62	972.87 ± 114.04	208.41 ± 12.01	25.13 ± 0.64
MetaGAT	165.18 ± 0.00	278.67 ± 0.00	379.47 ± 0.00	288.10 ± 0.00	90.41 ± 0.00	52.71 ± 0.00	241.51 ± 0.00	753.88 ± 0.00	177.10 ± 0.00	26.59 ± 0.00
Ours$_{GNN}$	164.46 ± 0.00	249.64 ± 0.00	366.97 ± 0.00	298.97 ± 0.00	91.02 ± 0.00	52.43 ± 0.00	194.36 ± 0.00	762.61 ± 0.00	178.69 ± 0.00	27.26 ± 0.00
Ours	**162.65 ± 0.00** (+1.5%)	**243.26 ± 6.49** (+12.9%)	**361.38 ± 0.00** (+5.1%)	**280.20 ± 0.00** (+2.7%)	**88.49 ± 0.00** (+2.1%)	**50.90 ± 0.89** (+3.4%)	**193.84 ± 0.00** (+19.7%)	**705.48 ± 0.00** (+6.4%)	**160.42 ± 0.00** (+9.6%)	**24.71 ± 0.00** (+7.1%)

Table 2: Performance when transferring to an unseen scenario, with the format as "mean ± standard deviation (gain in % compared with the best baseline)", the best **boldfaced** and second best underlined. We employ zero-shot transfer for evaluation: for each test scenario, we employ six other scenarios during the training process and then directly evaluate them in the respective test scenario. Our proposed method achieves the best performance in zero-shot transfer. (Some standard deviation = 0.00 because only two digits are kept.)

Methods	**Avg. Trip Time (seconds)**					**Avg. Delay Time (seconds)**				
	Grid4×4 (seen)	*Grid5×5 (seen)*	*Arterial4×4 (seen)*	*Ingolstadt21 (seen)*	*Cologne8 (seen)*	*Grid4×4 (seen)*	*Grid5×5 (seen)*	*Arterial4×4 (seen)*	*Ingolstadt21 (seen)*	*Cologne8 (seen)*
FTC	206.68 ± 0.54	550.38 ± 8.31	828.38 ± 8.17	319.41 ± 24.48	124.4 ± 1.99	94.64 ± 0.43	790.18 ± 7.96	1234.30 ± 6.50	183.70 ± 26.21	62.38 ± 2.95
MaxPressure	175.97 ± 0.70	274.15 ± 15.23	686.12 ± 9.57	375.25 ± 2.40	95.96 ± 1.11	64.01 ± 0.71	240.00 ± 18.43	952.53 ± 12.48	275.36 ± 14.38	31.93 ± 1.07
MPLight	179.51 ± 0.95	261.76 ± 6.60	541.29 ± 45.24	319.28 ± 10.48	98.44 ± 0.62	67.52 ± 0.97	213.78 ± 14.44	1083.18 ± 63.38	185.04 ± 10.70	34.38 ± 0.63
IPPO	167.62 ± 2.42	259.28 ± 9.55	431.31 ± 28.55	379.22 ± 34.03	90.87 ± 0.40	56.38 ± 1.46	243.58 ± 9.29	914.58 ± 36.90	247.68 ± 35.33	26.82 ± 0.43
rMAPPO	164.96 ± 1.87	300.90 ± 8.31	565.67 ± 44.8	453.61 ± 29.66	97.68 ± 2.03	53.65 ± 1.00	346.78 ± 28.25	1185.2 ± 167.48	372.2 ± 39.85	33.37 ± 1.97
CoLight	163.52 ± 0.00	242.37 ± 0.00	409.93 ± 0.00	337.46 ± 0.00	89.72 ± 0.00	51.58 ± 0.00	248.32 ± 0.00	776.61 ± 0.00	226.06 ± 0.00	25.56 ± 0.00
MetaLight	169.21 ± 1.26	247.83 ± 5.99	381.77 ± 12.85	292.26 ± 4.40	91.57 ± 0.75	57.56 ± 0.76	209.13 ± 19.40	862.32 ± 39.01	164.80 ± 3.75	27.61 ± 0.78
GESA	**161.33 ± 1.34**	252.11 ± 9.94	393.57 ± 13.72	320.02 ± 5.57	90.59 ± 0.74	**49.60 ± 0.71**	210.74 ± 13.56	775.22 ± 8.63	209.57 ± 3.32	26.50 ± 0.87
MetaGAT	165.23 ± 0.00	266.60 ± 0.00	374.80 ± 0.87	290.73 ± 0.45	90.74 ± 0.00	53.20 ± 0.00	234.80 ± 0.00	772.36 ± 0.00	176.86 ± 2.37	26.85 ± 0.00
Ours$_{GNN}$	164.32 ± 0.00	233.12 ± 0.00	382.38 ± 0.00	319.98 ± 0.00	89.29 ± 0.00	51.83 ± 0.00	204.24 ± 0.00	710.68 ± 0.00	184.10 ± 0.00	25.00 ± 0.00
Ours	162.47 ± 0.00 (-0.7%)	**220.63 ± 0.00** (+9.0%)	**349.60± 0.00** (+6.7%)	**278.05 ± 0.00** (+4.4%)	**88.55 ± 0.00** (+1.4%)	50.27 ± 0.00 (-1.3%)	**187.74 ± 0.00** (+11.0%)	**697.79 ± 0.00** (+9.7%)	**160.39 ± 0.00** (+2.7%)	**24.31 ± 0.00** (+4.9%)

Table 3: Performance on various scenarios that is seen in the training: methods from FTC to CoLight are trained and tested on the same scenario of each column, and methods from MetaLight to Ours are trained with all seven scenarios, five of them have results in each column.

4.4 Results

Great Transferability When Handling A New Scenario

We use a zero-shot way to evaluate each model's performance in transferring to new scenarios. Only transferable models are selected. In Table 2, each column means that we use the other six scenarios during training and then directly transfer to this unseen scenario. Our model achieves the best transfer results in all scenarios. **(1) Cooperation is needed:** Unlike single-agent methods like MetaLight, cooperation enhances transferability. **(2) TonT is better for Meta MARL:** compared to the second-best MetaGAT, ours achieved a **+7.91%** improvement on average and **+16.3%** in *Grid5×5*, mainly because our unified Transformer on Transformer design captures collaborators' o, a, r interdependency for better local cooperation, and global cross-scenario dynamics via multi-scenario co-training, respectively. Yet, MetaGAT only focuses on the neighbors' o interdependency and lacks a scenario-agnostic training scheme. *In some cases (Grid5×5, Ingostadt21), MetaGAT cannot beat MetaLight, which means that when cooperation is not as well designed as ours, it can even worsen the transferability when dealing with multi-scenarios.* **(3) Lower Transformer is better for cooperation:** ours$_{GNN}$ flattens o, a, r, thus losing the o, a, r interdependency, further highlighting the necessity of the Lower Transformer.

Enhanced Performance in Non-transfer Setting

In this non-transfer setting, as shown in Table 3, the scenario in each column is present in training: specifically, all meta-learning-based methods from MetaLight to Ours are trained with all seven scenarios, while other methods from FTC to CoLight are trained and tested on the same one scenario. Figure 3 further plots the top three methods' training process. **(1)**

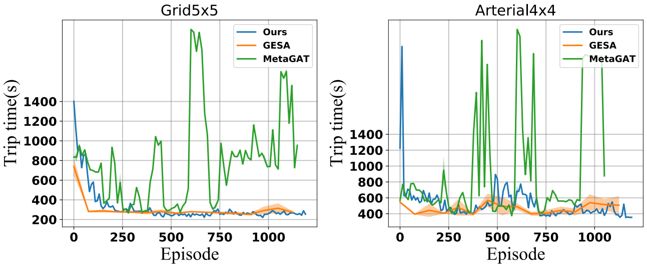


Figure 3: Trip time during the training process of Top 3 methods.

TonT is also advantageous for non-transfer: our method achieves the best results in four scenarios, with a **+6.20%** gain on average and **+9.96%** in *Grid5×5*. The gain is relatively lower than the Transfer setting, which is reasonable given we are dedicated to transfer. Besides, we have tiny **-1.02%** loss compared with the best GESA in the *Grid4×4*. A potential reason is the over-simplicity of the scenario. **(2) Our multi-scenario co-training is more stable than MetaGAT:** As shown in Figure 3, our training is much more stable than MetaGAT (both are multi-agent setting), due to the co-training design and strong scenario-agnostic trained Upper Transformer. GESA is the most stable given it is single-agent and also in a multi-scenario co-training scheme like ours.

Ablation Studies

The necessity of each component in model:

- **w/o Lower Transformer** replaces the Lower Transformer with MLP. No collaboration is formulated.
- **w/o Upper Transformer** replaces the Upper Transformer with GRU.

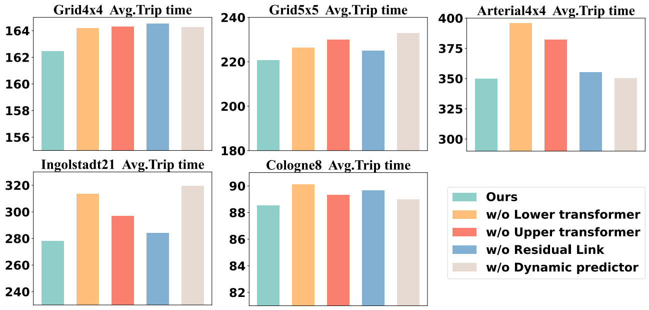


Figure 4: Ablation of each component in X-Light. The final solution attained the shortest trip time among all ablation experiments, showcasing the effectiveness of our design components.

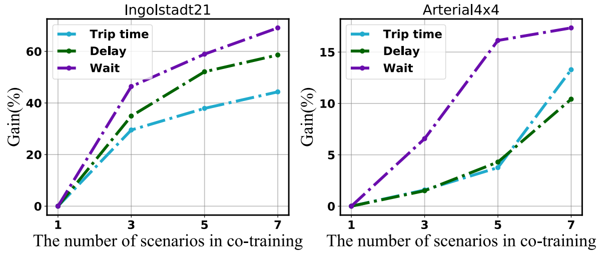


Figure 5: Impact of the number of scenarios co-training. As the number of scenarios increases, our performance also increases.

- **w/o Residual Link** removes the $\mathbf{o}_i^t$ in the actor input.
- **w/o Dynamic predictor** removes the dynamic predictor in the Upper Transformer, thus without $\mathcal{L}_{pred}$.

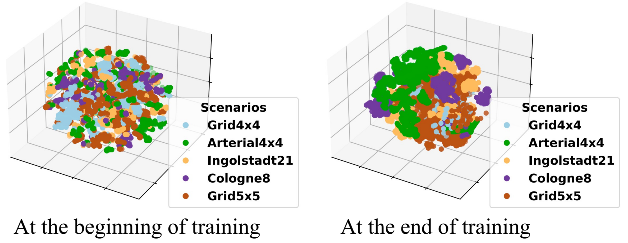
Figure 4 shows the trip time results, with other metrics in Appx. H. We can conclude that (1) the Lower Transformer, which promotes collaboration, is the most critical component. (2) In the Upper Transformer, the Transformer model is better than GRU; moreover, the dynamic predictor is also very necessary to ensure the Upper Transformer learns scenario-diagnostic dynamics. (3) As discussed before, the residual link can further boost performance, especially in easier scenarios such as *Grid4×4* and *Cologne8*, where the direct self-observation is more useful for decision in simple scenarios.

Numbers of co-trained scenarios: in Figure 5, we also conducted the impact of different numbers of co-training scenarios. We increase the number of co-trained scenarios from 1 to 7 and test in the two fixed scenarios (*Ingolstadt21* and *Arterial4×4*). Our method consistently improves as the number of co-training scenarios increases.

We also conducted experiments on the Upper Transformer, exploring various historical lengths, with details in Appx. H.

Visualizing X-Light's Upper and Lower Transformers

This visualization offers great insights on our TonT. **(1) Upper Transformer understands the scenario dynamics well**: We first analyze the output embeddings from the Upper Transformer. In Figure 6, we visualize the model before and after training in five evaluation scenarios with the 3D t-SNE technique, and each scenario has a unique color. The Upper Transformer can map dynamics from the same scenario together, demonstrating the Upper Transformer's good understanding of various scenario dynamics. **(2) Lower Transformer collaborates well by fully utilizing** o, a, r: Furthermore, we visualize the attention weights of the Lower Transformers. In Figure 7, we visualize attention weights for two heads across three layers in the *Cologne8*. We find that (1) in the shallow Layer-1, attention is mainly to observations (green) of the target and its neighbors via different heads; in deeper Layer-2, more attention is to actions (blue); in deepest Layer-3, right the output being used for decision, all o, r, a, especially rewards (grey), have high attention (more cases in Appx. I); (2) This further justify the necessity of our Lower Transformer, with superiority of modeling o, r, a-interrelation over GNN-based collaborations.



At the beginning of training

At the end of training

Figure 6: Visualizing Upper Transformer's output $\mathbf{z}^{\text{output}}$. Our model can group the decision dynamics from the same scenario together.

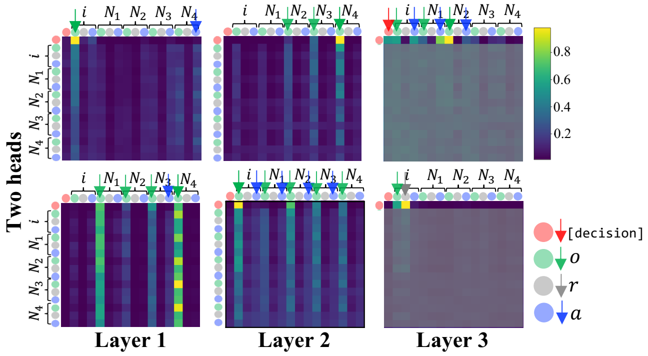


Figure 7: Visualizing Lower Transformer's attention (*Cologne8*), with top attentions marked with ↓. Each row in attention is the weight of the corresponding token w.r.t all other tokens. We pick 2 heads over 3 layers. The last layer exclusively uses the `[decision]` token (only the first row is shown). Lower Transformer efficiently models these MDP features' interrelations: more attention to o in shallow layers, and more to a, r in deeper layers.

Further Discussions: In Appx. A, we discuss the scalability, compatibility, assumptions for transferring, and theoretical guarantee of X-Light.

5 Conclusion

In this paper, we propose X-Light, an innovative framework that has two types of transformers with multi-scenario co-training. This design enhances the agent's transferability and enables collaboration across multiple intersections simultaneously. We use a large number of scenarios and comparison methods to verify the effectiveness of our proposed method. The outcomes highlight not only the exceptional performance of our method but also its robust transferability.

References

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

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

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

[Chu *et al.*, 2019] Tianshu Chu, Jie Wang, Lara Codecà, and Zhaojian Li. Multi-agent deep reinforcement learning for large-scale traffic signal control. *IEEE Transactions on Intelligent Transportation Systems*, 21(3):1086–1095, 2019.

[Dosovitskiy *et al.*, 2020] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[Du *et al.*, 2024] Xinqi Du, Ziyue Li, Cheng Long, Yongheng Xing, S Yu Philip, and Hechang Chen. Felight: Fairness-aware traffic signal control via sample-efficient reinforcement learning. *IEEE Transactions on Knowledge and Data Engineering*, 2024.

[Duan *et al.*, 2017] Yan Duan, John Schulman, Xi Chen, Peter L. Bartlett, Ilya Sutskever, and Pieter Abbeel. RL^2: Fast reinforcement learning via slow reinforcement learning, 2017.

[Goodfellow *et al.*, 2014] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

[Guo *et al.*, 2014] Haifeng Guo, Jun Cheng, Qitao Peng, Chao Zhu, and Yuanjie Mu. Dynamic division of traffic control sub-area methods based on the similarity of adjacent intersections. In *17th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, pages 2208–2213. IEEE, 2014.

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

[Hamilton *et al.*, 2017] Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. *Advances in neural information processing systems*, 30, 2017.

[Han *et al.*, 2021] Kai Han, An Xiao, Enhua Wu, Jianyuan Guo, Chunjing Xu, and Yunhe Wang. Transformer in transformer. *Advances in Neural Information Processing Systems*, 34:15908–15919, 2021.

[Humplik *et al.*, 2019] Jan Humplik, Alexandre Galashov, Leonard Hasenclever, Pedro A Ortega, Yee Whye Teh, and Nicolas Heess. Meta reinforcement learning as task inference. *arXiv preprint arXiv:1905.06424*, 2019.

[Hunt *et al.*, 1982] PB Hunt, DI Robertson, RD Bretherton, and M Cr Royle. The SCOOT on-line traffic signal optimisation technique. *Traffic Engineering & Control*, 23(4), 1982.

[Janner *et al.*, 2021] Michael Janner, Qiyang Li, and Sergey Levine. Offline reinforcement learning as one big sequence modeling problem. *Advances in neural information processing systems*, 34:1273–1286, 2021.

[Jiang *et al.*, 2024] Haoyuan Jiang, Ziyue Li, Zhishuai Li, Lei Bai, Hangyu Mao, Wolfgang Ketter, and Rui Zhao. A general scenario-agnostic reinforcement learning for traffic signal control. *IEEE Transactions on Intelligent Transportation Systems*, 2024.

[Kouvelas *et al.*, 2014] Anastasios Kouvelas, Jennie Lioris, S Alireza Fayazi, and Pravin Varaiya. Maximum pressure controller for stabilizing queues in signalized arterial networks. *Transportation Research Record*, 2421(1):133–141, 2014.

[Liang *et al.*, 2019] Xiaoyuan Liang, Xunsheng Du, Guiling Wang, and Zhu Han. A deep reinforcement learning network for traffic light cycle control. *IEEE Transactions on Vehicular Technology*, 68(2):1243–1253, 2019.

[Liang *et al.*, 2022] Yuxuan Liang, Pan Zhou, Roger Zimmermann, and Shuicheng Yan. Dualformer: Local-global stratified transformer for efficient video recognition. In Shai Avidan, Gabriel Brostow, Moustapha Cissé, Giovanni Maria Farinella, and Tal Hassner, editors, *Computer Vision – ECCV 2022*, pages 577–595, Cham, 2022. Springer Nature Switzerland.

[Lou *et al.*, 2022] Yican Lou, Jia Wu, and Yunchuan Ran. Meta-reinforcement learning for multiple traffic signals control. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pages 4264–4268, 2022.

[Lowrie, 1990] PR Lowrie. SCATS, Sydney co-ordinated adaptive traffic system: A traffic responsive method of controlling urban traffic. 1990.

[Lu *et al.*, 2023] Jiaming Lu, Jingqing Ruan, Haoyuan Jiang, Ziyue Li, Hangyu Mao, and Rui Zhao. Dualight: Enhancing traffic signal control by leveraging scenario-specific and scenario-shared knowledge. *arXiv preprint arXiv:2312.14532*, 2023.

[Mao *et al.*, 2023] Hangyu Mao, Rui Zhao, Ziyue Li, Zhiwei Xu, Hao Chen, Yiqun Chen, Bin Zhang, Zhen Xiao, Junge Zhang, and Jiangjin Yin. Pdit: Interleaving perception and decision-making transformers for deep reinforcement learning. *arXiv preprint arXiv:2312.15863*, 2023.

[Melo, 2022] Luckeciano C Melo. Transformers are meta-reinforcement learners. In *international conference on machine learning*, pages 15340–15359. PMLR, 2022.

[Oroojlooy *et al.*, 2020] Afshin Oroojlooy, Mohammadreza Nazari, Davood Hajinezhad, and Jorge Silva. Attendlight: Universal attention-based reinforcement learning model for traffic signal control. *Advances in Neural Information Processing Systems*, 33:4079–4090, 2020.

[Prashanth and Bhatnagar, 2011] LA Prashanth and Shalabh Bhatnagar. Reinforcement learning with average cost for adaptive control of traffic lights at intersections. In *2011 14th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, pages 1640–1645. IEEE, 2011.

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

[Rostami *et al.*, 2020] Mohammad Rostami, David Isele, and Eric Eaton. Using task descriptions in lifelong machine learning for improved performance and zero-shot transfer. *Journal of Artificial Intelligence Research*, 67:673–704, 2020.

[Schulman *et al.*, 2017] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[Smith *et al.*, 2013] Stephen F Smith, Gregory Barlow, Xiao-Feng Xie, and Zachary B Rubinstein. Surtrac: Scalable urban traffic control. 2013.

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

[Wei *et al.*, 2019a] Hua Wei, Chacha Chen, Guanjie Zheng, Kan Wu, Vikash Gayah, Kai Xu, and Zhenhui Li. Presslight: Learning max pressure control to coordinate traffic signals in arterial network. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1290–1298, 2019.

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

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

[Wolshon *et al.*, 2016] Brian Wolshon, Anurag Pande, et al. *Traffic engineering handbook*. John Wiley & Sons, 2016.

[Wu *et al.*, 2023] Qiang Wu, Mingyuan Li, Jun Shen, Linyuan Lü, Bo Du, and Ke Zhang. 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*, KDD '23, page 2639–2647, New York, NY, USA, 2023. Association for Computing Machinery.

[Xu *et al.*, 2013] Lun-Hui Xu, Xin-Hai Xia, and Qiang Luo. The study of reinforcement learning for traffic self-adaptive control under multiagent markov game environment. *Mathematical Problems in Engineering*, 2013, 2013.

[Xu *et al.*, 2022] Mengdi Xu, Yikang Shen, Shun Zhang, Yuchen Lu, Ding Zhao, Joshua Tenenbaum, and Chuang Gan. Prompting decision transformer for few-shot policy generalization. In *international conference on machine learning*, pages 24631–24645. PMLR, 2022.

[Yu *et al.*, 2022a] Chao Yu, Akash Velu, Eugene Vinitsky, Jiaxuan Gao, Yu Wang, Alexandre Bayen, and Yi Wu. The surprising effectiveness of ppo in cooperative multi-agent games. *Advances in Neural Information Processing Systems*, 35:24611–24624, 2022.

[Yu *et al.*, 2022b] Rui Yu, Dawei Du, Rodney LaLonde, Daniel Davila, Christopher Funk, Anthony Hoogs, and Brian Clipp. Cascade transformers for end-to-end person search. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7267–7276, June 2022.

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

[Zhang *et al.*, 2020] Huichu Zhang, Chang Liu, Weinan Zhang, Guanjie Zheng, and Yong Yu. Generalight: Improving environment generalization of traffic signal control via meta reinforcement learning. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, pages 1783–1792, 2020.

[Zheng *et al.*, 2019] Guanjie Zheng, Yuanhao Xiong, Xinshi Zang, Jie Feng, Hua Wei, Huichu Zhang, Yong Li, Kai Xu, and Zhenhui Li. Learning phase competition for traffic signal control. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pages 1963–1972, 2019.

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