EUR Research Information Portal

A General Scenario-Agnostic Reinforcement Learning for Traffic Signal Control

Published in:
IEEE Transactions on Intelligent Transportation Systems

Publication status and date:
Published: 27/03/2024

DOI (link to publisher):
10.1109/TITS.2024.3377106

Document Version
Publisher's PDF, also known as Version of record

Document License/Available under:
Article 25fa Dutch Copyright Act

Citation for the published version (APA):
Jiang, H., Li, Z., Li, Z., Bai, L., Mao, H., Ketter, W., & Zhao, R. (2024). A General Scenario-Agnostic Reinforcement Learning for Traffic Signal Control. *IEEE Transactions on Intelligent Transportation Systems*, *25*(9), 11330-11344. https://doi.org/10.1109/TITS.2024.3377106

Link to publication on the EUR Research Information Portal

Terms and Conditions of Use
Except as permitted by the applicable copyright law, you may not reproduce or make this material available to any third party without the prior written permission from the copyright holder(s). Copyright law allows the following uses of this material without prior permission:

- you may download, save and print a copy of this material for your personal use only;
- you may share the EUR portal link to this material.

In case the material is published with an open access license (e.g. a Creative Commons (CC) license), other uses may be allowed. Please check the terms and conditions of the specific license.

Take-down policy
If you believe that this material infringes your copyright and/or any other intellectual property rights, you may request its removal by contacting us at the following email address: openaccess.library@eur.nl. Please provide us with all the relevant information, including the reasons why you believe any of your rights have been infringed. In case of a legitimate complaint, we will make the material inaccessible and/or remove it from the website.

Erasmus University Rotterdam
Making Minds Matter

A General Scenario-Agnostic Reinforcement Learning for Traffic Signal Control

Haoyuan Jiang, Ziyue Li, Zhishuai Li, Lei Bai, Hangyu Mao, Wolfgang Ketter, and Rui Zhao

***Abstract*— Reinforcement learning (RL) can automatically learn a better policy through a trial-and-error paradigm and has been adopted to revolutionize and optimize traditional traffic signal control systems that are usually based on handcrafted methods. However, most existing RL-based models are either based on a single scenario or multiple independent scenarios, where each scenario has a separate simulation environment with predefined road network topology and traffic signal settings. These models implement training and testing in the same scenario, thus being strictly tied up with the specific setting and sacrificing model generalization heavily. While a few recent models could be trained by multiple scenarios, they require a huge amount of manual labor to label the intersection structure, hindering the model's generalization. In this work, we aim at a *general* framework that could eliminate heavy labeling and model a variety of scenarios *simultaneously*. To this end, we propose a general Scenario-Agnostic (GESA) reinforcement learning framework for traffic signal control with: (1) A general plug-in module to map all different intersections into a unified structure, freeing us from the heavy manual labor to specify the structure of intersections; (2) A unified state and action space design to keep the model input and output consistently structured; (3) A large-scale co-training with multiple scenarios, leading to a generic traffic signal control algorithm. GESA can automatically handle various structured intersections from various cities without human labeling, and it co-trains a generalist agent to control traffic signals for multiple cities together, which also demonstrates superior transferability in zero-shot settings. In experiments, we demonstrate our algorithm as the first one that can be co-trained with seven different scenarios without manual annotation and gets 13.27% higher rewards than baselines. When dealing with a new scenario, our model can still achieve 9.39% higher rewards. The code, scenarios, and demos are available here.**

***Index Terms*— General model, reinforcement learning, traffic control.**

I. Introduction

AN EFFICIENT traffic signal control (TSC) system is undoubtedly a cost-effective means of alleviating congestion [1], [2]. Traditional TSC systems, such as SCOOT [3] and SCATS [4], are widely deployed in hundreds of cities worldwide, yet their traffic signal plans are manually designed according to rules from expert knowledge and transportation engineering. For example, Webster's Equation [5], one of the most widely-used methods, directly uses the traffic volumes to calculate the optimal cycle length and phase split for a single intersection to minimize the total travel time at this intersection. However, those methods may be powerless when handling dynamic traffic networks without well-observed traffic flow patterns. To this end, reinforcement learning (RL) [6], [7], [8] has been preferably adopted into the TSC domain since it is a learning-based method with higher automation. Such a trial-and-error paradigm based on the traffic simulator has demonstrated better performance than transport engineering-based methods [9].

The recent RL-based TSC models can be roughly divided into two categories based on the scenarios where the training and testing are conducted. A scenario is usually a simulation environment that contains a set of intersections: (1) **Single-scenario RL**, as the majority, its training and testing need to be on the same scenario [6], [10], [11], [12], [13], [14]. However, the model will be unusable or perform badly in a new scenario. For example, in Fig. 1(b) top, these methods might be trained and tested in the same scenario with 5×5 four-approach intersections of Fig. 1(a2), but these methods will ill-perform in another new scenario with mixed intersections of Fig. 1(a1) and 1(a2). (2) **Multi-scenario RL**, as shown in Fig. 1(b) bottom, where training is conducted in multiple scenarios, and testing could be in different scenarios. For example, [15], [16], and [17] are proposed to train a TSC system with multi-scenarios. However, in the training stage, the existing multi-scenario RL models need heavy manual labor to annotate the structure of intersections, such as the direction of each entering approach, the number of entering lanes of each entering approach, the traffic movement of each entering lane, etc. Moreover, they either achieve multi-scenario co-training in a sequential manner, one scenario after another, leading to a rather unstable learning curve and slower convergence [15], [17] or only narrow the scale of a scenario to only one intersection in one scenario, which heavily limits the model's generalization [16].

Moreover, current RL-based TSC methods are trained with several pre-defined and fixed scenarios, whereas they cannot gain generalization capability without labeling, which limits the application of RL-based methods in the real world. These methods can exploit the various traffic flows generated by the simulator to make the model effective in training scenarios, but finding a low-cost universal method with promising transferability meanwhile is still a research gap.

Manuscript received 4 May 2023; revised 9 December 2023; accepted 24 February 2024. Date of publication 27 March 2024; date of current version 29 August 2024. The Associate Editor for this article was M. Nicoli. *(Corresponding author: Ziyue Li.)*

Haoyuan Jiang, Zhishuai Li, Hangyu Mao, and Rui Zhao are with SenseTime Research, Beijing 100080, China.

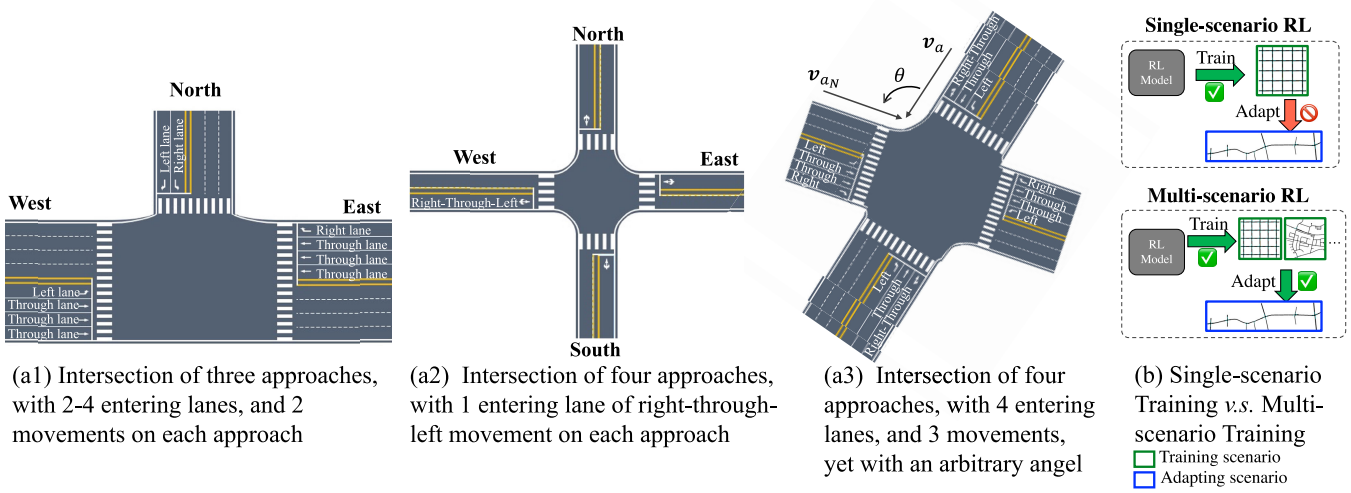
Ziyue Li and Wolfgang Ketter are with the Department of Information Systems, University of Cologne, 50923 Köln, Germany (e-mail: zlibn@wiso.uni-koeln.de).

Lei Bai is with Shanghai AI Laboratory, Shanghai 200232, China.

Digital Object Identifier 10.1109/TITS.2024.3377106

1558-0016 © 2024 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See https://www.ieee.org/publications/rights/index.html for more information.

Authorized licensed use limited to: Erasmus Universiteit Rotterdam. Downloaded on February 25,2025 at 07:07:54 UTC from IEEE Xplore. Restrictions apply.



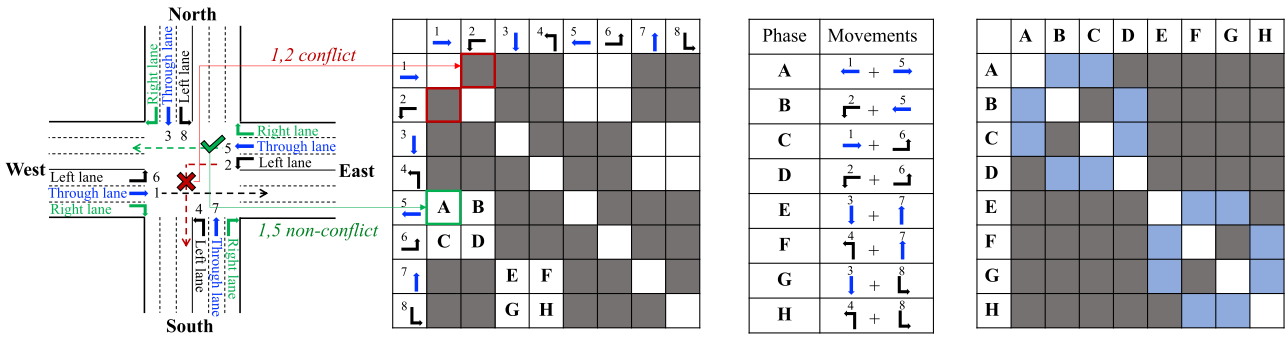
(a1) Intersection of three approaches, with 2-4 entering lanes, and 2 movements on each approach

(a2) Intersection of four approaches, with 1 entering lane of right-through-left movement on each approach

(a3) Intersection of four approaches, with 4 entering lanes, and 3 movements, yet with an arbitrary angel

(b) Single-scenario Training *v.s.* Multi-scenario Training

Fig. 1. (a) Three intersections with different structures in terms of approach, movement, and lane; (b) Single-scenario RL *v.s.* Multi-scenario RL.



(a) A standard intersection with four approaches and three lanes (right, through, left) within each approach

(b) Conflict matrix for eight movements, with eight available phases A-H and conflicts in grey.

(c) Eight phases from non-conflicting movements

(d) Phase Competition Mask: Grey: competing and conflicting; Blue: partial competing

Fig. 2. A standard intersection structure, conflict matrix, and available traffic signal phases.

As a result, the existing methods still face tremendous challenges in jumping out from the simulation and implementing them in real cities. This is known as *sim2real* challenge. The challenges mainly come from the wide gap between the real complex cities and the simplified simulation systems. In the real world, the intersection structure could be rather versatile in terms of different settings of approaches (*i.e.*, north, south, east, west), movements (*i.e.*, left, right, through), and lanes (*e.g.*, two through lanes, one right-through lane). As shown in Fig. 1(a), an intersection could have a different amount of approaches. Within an approach, there can be different combinations of movements ; A lane could also combine different movements. However, most of the existing methods only consider a standard simulation intersection with four approaches and three lanes (right, through, and left) within each approach, shown in Fig. 2(a). This largely limits the model generalization.

To conclude, a qualified TSC approach needs high generalization and effectiveness: it should handle various intersections and be able to transfer to other unseen targets easily and with low cost. In this paper, we aim to answer three questions: (1) How do we co-train an RL with multi-scenarios without labeling, given the diverse intersection structures? (2) Will multi-scenario co-training improve the TSC? If yes, why? And the more scenarios, the better? (3) Does the co-trained RL model still perform well in the new scenario?

To narrow the *sim2real* gap significantly and get more ready to be deployed in real cities, in this paper, we provide a **GE**neral **S**cenario-**A**gnostic (GESA) reinforcement learning framework for the TSC task. To our best knowledge, GESA is the first work that pursues high generability and co-trains multiple scenarios without labels: it automatically handles various scenarios; the reinforcement learning is designed accordingly to achieve generalization; it is co-trained with multiple scenarios simultaneously and demonstrates high transferability. Specifically, to co-train in multiple scenarios with various intersections, the vectors with approach spatial information are employed to map shape-odd and complex intersections into the standard intersection. Then, the mapped intersections are used to generate the characteristic information of each traffic movement and the phase of the traffic lights in a specific order. Finally, we extend the original FRAP [11] to a policy gradient-based framework, which can facilitate the model coverage and is compatible with different intersections.

Authorized licensed use limited to: Erasmus Universiteit Rotterdam. Downloaded on February 25,2025 at 07:07:54 UTC from IEEE Xplore. Restrictions apply.

The contributions are summarized in three-fold: (1) We present a general plug-in module to map the intersections into a unified structure, freeing us from the heavy manual labeling work to specify the intersection structure and enabling large-scale co-training under multiple different scenarios. (2) Accordingly, we design a unified state and action space to keep the model input and output structure consistent for more general capabilities. Moreover, the GESA can adapt to various unseen scenarios and achieve promising performance without re-training. (3) We build two real-world scenarios using the real city road map and the real traffic dynamics, together with five public scenarios, where we co-train and validate the GESA with prudent experiments. All these lead us closer to the ultimate goal: to implement RL-based TSC in real cities.

The rest of the paper is organized as follows. In Section II, we review related work for TSC. Section III provides the preliminaries, and Section IV introduces the model thoroughly. Experiments are given in Section V. The conclusion is presented in Section VI.

II. Related Work

We will review the existing methods for TSC from the two methodology stakes: transportation engineering-based methods and reinforcement learning-based methods.

A. Traditional Traffic Signal Control

Traditional methods are widely used in the real world. Most of them depend on strong assumptions and manually specified rules. The fixed-time [18] method decided the traffic signal plan as a fixed cycle length, which is not flexible enough and cannot automatically adapt its logic to the actual situation. Actuated control model [19], [20] instead used pre-defined sets of rules to decide whether to adjust the current phase. SCOOT [3] and SCATS [4] pre-defined several signal plans and selected the best of them according to the internal measurements. Recently, MaxPressure (MP) [21], [22] was designed to optimize vehicle travel time and achieve expressive performance dynamically. It balances the queue length between incoming and outgoing lanes to reduce the risk of over-saturation. This way, its logic can be dynamically changed according to the situation. Reference [23] further proposed an Advanced MP to consider running and queueing vehicles. All these methods are simple and interpretable and can be applied to most traffic signal systems in the real world, but their assumption dependencies are relatively simple and rigid. Actual situations may not be satisfied with them, so traditional methods usually cannot guarantee optimal results.

B. Reinforcement Learning-Based Traffic Signal Control

Recently, reinforcement learning methods have been deeply researched for the TSC task [1], [24] and other mobilities [25]. In [26] and [27], all intersections in a scenario are controlled by a global agent, where the state and action space are joint. The methods in [6], [10], and [28] handle the states separately, but they are still modeled as a joint action. These methods cannot be trained only once and then directly transferred to a new scenario. Thus, if one needs to adapt the well-trained model to an unseen intersection, model re-training in this new scenario is necessary. Such approaches have extended time and money costs, and their performance cannot be guaranteed. In recent years, independent RL has become popular because of its low computational cost, feasibility, and easy scale-up. It is more suitable for application in the real world. IntelliLight *etc.* [7], [29] treated the states as an image for each intersection and handled actions independently. NFQI [30], CoLight [31], and QCOMBO [12] introduce communication to coordinate the actions between intersections. FRAP [11] presents a flexible network FRAP (short for Flipping and Rotation and considering All Phase configurations) that can be used with different intersections based on phase competition. Based on FRAP, MPLight [13] add the pressure observation into the states and rewards to extend FRAP to a thousand-light control scenario. Then, MTLight [14] introduces a multi-task pipeline, with an additional task to predict next step states, which can express more general underlying characteristics. However, all these methods need to be trained and evaluated under the same scenarios to get better performance, which limits the development of the algorithm from simulation to reality.

C. More Generalized RL for Traffic Signal Control

More generalized RL agents are proposed for traffic signal control [15], [16], [17]. AttendLight [16] introduced a training framework suitable for intersections of varying traffic flows and configurations by utilizing two attention models. GeneraLight [32] uses GAN [33] to generate traffic diversity, thereby enhancing the robustness of different traffic flows. However, these two methods increase the generalizability toward various flow patterns instead of various structured scenarios; thus, each scenario of these two methods only considers one intersection.

To fully generalize to multi-scenario, meta-reinforcement learning is incorporated. MetaVIM [34] uses context-based Meta RL to solve the generalization problem and uses intrinsic rewards to cooperate. Still, it is only trained in one scenario, so it cannot be transferred to an environment that is different from the training scenario. The most related work is MetaLight [15], which uses gradient-based meta-reinforcement learning and multiple scenarios for training. However, all scenarios used for training in MetaLight are rather homogeneous, which heavily limits the generalization. Moreover, all of them still need heavy manual work to label the intersection structure, and their training mechanisms (feeding one scenario after another) introduce significant instability).

In this paper, to overcome these limitations, we design a flexible plug-in module to automatically process the various structures of intersections and extend the FRAP model to work in a unified state and action space, making it possible to co-train with multi-scenarios without labeling needed.

III. Preliminaries

In this paper, the TSC is conducted at a single intersection. As mentioned before, several intersections constitute a scenario. In the following, the problem of this work is formulated

Authorized licensed use limited to: Erasmus Universiteit Rotterdam. Downloaded on February 25,2025 at 07:07:54 UTC from IEEE Xplore. Restrictions apply.

formally, and then, for a particular intersection, the adopted terms are declared.

Problem Statement: The TSC is a sequential decision task characterized by the Markov decision process (MDP), which can be formulated as a single-agent RL problem $< \mathcal{S}, \mathcal{A}, \mathcal{R}, \gamma, \pi >$. Given the state set $\mathcal{S}$, which includes all related information such as traffic volume and current phase, action set $\mathcal{A}$, the reward function $\mathcal{R}$, and discount factor γ, at each intersection i and each timestep t the goal of the agent is to find a policy $\pi(a|s)$ that maximizes the expected return $G^t := \sum_{l=0}^{\infty} \gamma^l r^{t+l}$.

Definition 1 (Intersection): The types of intersections are versatile in the real world. A standard intersection is shown in Fig. 2(a), and the others can be treated as a variant of it. Each intersection has several approaches, where approaches can be categorized as the entering approach and the exiting approach. There are four entering approaches in Fig. 2(a) with lanes labeled, located in east, west, north, and south directions, respectively. There are also four existing approaches in Fig. 2(a), which are left blank. In the real world, most intersections consist of three or four approaches.

Definition 2 (Entering Approach): In each entering approach, there should be at least one entering lane. In Fig. 2(a), each approach has three entering lanes: right, through, and left.

Definition 3 (Traffic Movement): A traffic movement is defined as a vehicle moving from an entering approach to an exiting approach. Fig. 2(a) shows eight movements (left and through) that are signal-controlled and four right movements that are free (since right lanes do not conflict with other lanes). The conflict matrix of the eight movements is shown in Fig. 2(b), where a grey entry means the two movements cannot be released at the same time due to conflict. A movement can have more than one entering lane, and an entering approach can also contain several movements.

Definition 4 (Phase): A traffic phase is a combination of movement signals. As illustrated in Fig. 2(c), there are eight phases in the standard intersection, and each phase combines two traffic movements without conflicts. Note that not all the phases are available for the actual intersections. The phase competition matrix is shown in Fig. 2(d), where a grey entry means two phases cannot be released due to the conflicts from their belonging movements, and a blue entry means partial conflict since two phases share one same movement.

IV. Method

In this section, the details of the proposed GESA model will be given. Section IV-A and Section IV-B will introduce (1) the well-elaborated plug-in module that unifies all intersections and (2) the unified state and action design, respectively, in a general way. Section IV-C will give setting details for RL.

A. Toward to a Unified Intersection Structure

In contrast to the standard intersection in Fig. 2(a), the structures of the real-world intersections are arbitrary, even shape-odd. Fig. 1(a1) shows a three-approach intersection. Compared with the standard intersection, its south approach and the through lane in the north approach are absent. In Fig. 1(a2), each entering lane has multiple traffic movements, which are often observed in urban streets. The approach directions are irregular in Fig. 1(a3), which is also common.

To unify different intersections, the General Plug-In(GPI) is designed, which maps the intersections with arbitrary structures into the standard one, making each entering lane only have one main movement.

1) To Unify the Approaches: Essentially, the unification of approaches is to align their directions in the target intersection and the standard intersection. We provide the approach-unifying process, shown in Algorithm 1. For particular intersection i, we first retrieve the position vectors (e.g., geo-locations) for the approaches and count their numbers. Then, the position vectors are employed to calculate the angles between the approaches: The angle $\theta_{a,a'}$ between approaches a and a' is calculated by $\theta_{a,a'} = \arccos(\frac{\mathbf{v}_a \cdot \mathbf{v}_{a'}}{\|\mathbf{v}_a\| \cdot \|\mathbf{v}_{a'}\|})$, where $\mathbf{v}_a$ and $\mathbf{v}_{a'}$ represent the spatial locations of approaches a and a', respectively. Finally, the intersection is matched to the standard one according to the angles, as shown in Fig. 1(a3): (1) An approach is selected to associate with the north approach a_N in the standard intersection. (2) The other approaches $\{a\}$s, denoted as $\mathbf{a}$, are matched approximately to the standard intersection's approaches according to the related angles. Specifically, if the angle between the a_N and a is between 45° and 135°, a will be matched to the east approach a_E. South and west approaches are matched similarly. If each direction matches no more than one approach, the matching process will be successful.

2) To Unify the Movements: The objective of movement unification is to achieve one-to-one matching between lanes and movements and handle unavailable phases. For the case that one lane is associated with multiple movements (e.g., Fig. 1(a2)), we can delineate their priority according to the traffic rules [5]: (1) Through movement is with the highest priority. (2) Turning left has priority over turning right. (3) Turning right commonly is free and needs no signal control. Thus, we can take all entering lanes as through lanes in Fig. 1 (a2). Finally, eight traffic signal phases can be generated, as shown in Fig. 2 (c). Red signals are adopted to fill the unavailable phases.

All the above rules are integrated into a plug-in module, which can be placed in the simulation scenarios' initialization stage while eschewing extensive manual labor to label the distinct intersections.

B. Toward to a General Model

To ensure that the RL model gains better transferability, it should be compatible with different kinds of intersections. Essentially, the in-out dimensions of the model need to be constant regardless of the intersections. Thus, it is crucial to unify the agent's state space and action space.

1) A General State Space: For the agent state space, with no need to control the right-turn behaviors, we first gather the observations with eight groups according to the movements they belong in the standard intersection (as Fig. 2(a)). There are multiple demands for each group, such as queue length,

Authorized licensed use limited to: Erasmus Universiteit Rotterdam. Downloaded on February 25,2025 at 07:07:54 UTC from IEEE Xplore. Restrictions apply.

Algorithm 1 Algorithm to Unify the Approach Directions

Input: The number of approaches of target intersection $|\mathbf{a}|$; The position vectors of the approaches $\mathbf{V} = \{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_{|\mathbf{a}|}\}$;

Initialization: Indicator of success in unification *Flag=False*;
for a^* in $\{a\}$s **do**
  Initialization: Approach-direction set $\mathcal{M} = \{m_N : \text{set}(a^*), m_S : \text{set}(), m_W : \text{set}(), m_E : \text{set}()\}$;
  for a in $\{a\}$s and $a \neq a^*$ **do**
    $\theta_{a,a^*} = \arccos(\frac{\mathbf{v}_a \cdot \mathbf{v}_{a^*}}{\|\mathbf{v}_a\| \cdot \|\mathbf{v}_{a^*}\|})$
    if $\mathbf{v}_a \times \mathbf{v}_{a^*} < 0$ **then**
      $\theta_{a,a^*} = 360° - \theta_{a,a^*}$
    end if
    if $45° \leq \theta_{a,a^*} < 135°$ **then**
      m_E.add(a)
    else if $135° \leq \theta_{a,a^*} < 225°$ **then**
      m_S.add(a)
    else if $225° \leq \theta_{a,a^*} < 315°$ **then**
      m_W.add(a)
    else
      m_N.add(a)
    end if
  end for
  if length(m_N)<2 and length(m_S)<2 and length(m_W)<2 and length(m_E)<2 **then**
    Flag=True;
    Break;
  end if
end for

Output: *Flag*, $\mathcal{M}$.

flow, and occupancy. Second, the movements of multiple entry lanes may be the same. In this case, we aggregate the average observations in the lanes to represent the demand for the movement. Third, considering that some traffic movements may be absent compared with the standard intersection, we use zero padding in the missing group demands and add a binary indicator in the corresponding states.

2) A General Action Space: The agent action is to choose the phase for the next time interval. For the action space, since the structure of intersections is already unified, we reserve all eight phases, and the order of these phases is fixed (same as Fig. 2(c)), letting the agent have the same logic to select phases. However, not all eight phases are always available, e.g., phases E and F in Fig. 2(c) do not appear in the intersection in Fig. 1(a1). To solve this, we use mapped intersection structures to identify unavailable phases and add a mask to let the agent ignore them.

3) Learning: Our unified design can enable most RL models to handle arbitrary intersections. For instance, if adopting our unified design, a generic Proximal Policy Optimization (PPO) [35] can be co-trained with multiple scenarios by asynchronous learning [36].

We modify the FRAP to a policy gradient-based design. Our framework is shown in Fig. 3. We denote in the i-th intersection from the k-th scenario, the j-th movement's n-th feature as $s^m_{i,j,k,n}$, where $i \in \mathbb{R}^{I_k}$, $k \in [1, \ldots, K]$, and $j \in [1, \ldots, 8]$ as eight movements.

Firstly, via the GPI module, arbitrary intersections from various scenarios are mapped into the standard one. Thus, the feature will be scenario-agnostic and intersection-agnostic.

$$\tilde{s}^m_{i,j,n} = GPI(s^m_{i,j,k,n}) \tag{1}$$

Then, the embedding for the j-th traffic movement's all features is obtained by Multi-Layer Perceptron (MLP) and concatenation (denoted as operator $\|$), as shown in Fig. 4. Besides, the last feature "indicator" (binary) denotes whether this movement exists in this intersection.

$$\bar{\mathbf{s}}^m_{i,j} = \|_{n=1}^{N+1} MLP(\tilde{s}^m_{i,j,n}) \tag{2}$$

Secondly, the calculated movement embedding $\bar{\mathbf{s}}^m_{i,j}$ is fed into the FRAP module. (1) Add: FRAP adds the embedding of movements as a phase representation. For the l-th phase which consists of movements $j, j', l \in [A, B, \ldots, H]$: $\mathbf{s}^p_{i,l} = \bar{\mathbf{s}}^m_{i,j} + \bar{\mathbf{s}}^m_{i,j'}$, where the two movements $j, j' \in [1, 2, \ldots, 8]$. (2) Phase-pair representation: Once obtaining the phase embedding, a phase-pair representation is constructed to capture the pairwise relations for competition: $\mathbf{s}^{pp}_i = \mathbf{s}^p_{i,l} \| \mathbf{s}^p_{i,l'}$. (3) Phase competition: To avoid phase conflict, pairwise competition scores [11] are learned as a competition mask, denoted as $\mathbf{\Omega}$. Thus, the phase-pair representations $\mathbf{s}^{pp}_i$ will go through 1×1 convolution and multiply with the phase competition mask to yield the masked phase-pair embedding: $\mathbf{s}^{pp}_{masked\,i} = Conv_{1\times1}(\mathbf{s}^{pp}_i) \otimes \mathbf{\Omega}$, and $\otimes$ is Kronecker product. We denote the whole FRAP module as $FRAP(\cdot)$:

$$\mathbf{s}^{pp}_{masked\,i} = FRAP(\bar{\mathbf{s}}^m_{i,j}) \tag{3}$$

Thirdly, the final policy $\pi(\cdot)$ and value $V(\cdot)$ estimation networks are implemented as the actor and critic, which are conducted by MLP, respectively. In contrast to the original FRAP, which uses Deep Q-leaning (DQN) [37] as the action generator, the introduced actor-critic model can handle the action-continuous scenarios and is more friendly for the parallel training by A3C. To get the final action, the phase can be selected by the highest probability or sampled by the probability distribution of the i-th intersection for exploration from $\pi(a_i|s_i)$, where $\mathbf{a}_i \in \mathbb{R}^8$ for 8 phases.

$$\begin{aligned} &Actor:\ \pi(a_i|s_i) = Softmax\left(MLP\left(\sum(\mathbf{s}^{pp}_{actor\,i})\right)\right), \\ &\text{where } \mathbf{s}^{pp}_{actor\,i} = Conv_{1\times1}(\mathbf{s}^{pp}_{masked\,i}) \end{aligned} \tag{4}$$

$$\begin{aligned} &Critic:\ V(s_i) = MLP\left(\sum(\mathbf{s}^{pp}_{critic\,i})\right), \\ &\text{where } \mathbf{s}^{pp}_{critic\,i} = Conv_{1\times1}(\mathbf{s}^{pp}_{masked\,i}) \end{aligned} \tag{5}$$

Overall, compared with the original FRAP, our main differences are three-fold: (1) A general plug-in module, as well as the unified state and action, are designed. Such unified state space and action space enable the model to be co-trained with multi-scenarios and to handle several structure-distinct intersections simultaneously. (2) All eight movements are utilized as the state input at the same time, and each movement state is featured by traffic indexes and indicator embedding. In this way, the proposed framework gains better generalizability, and various independent RL-based models can be transferred

Authorized licensed use limited to: Erasmus Universiteit Rotterdam. Downloaded on February 25,2025 at 07:07:54 UTC from IEEE Xplore. Restrictions apply.

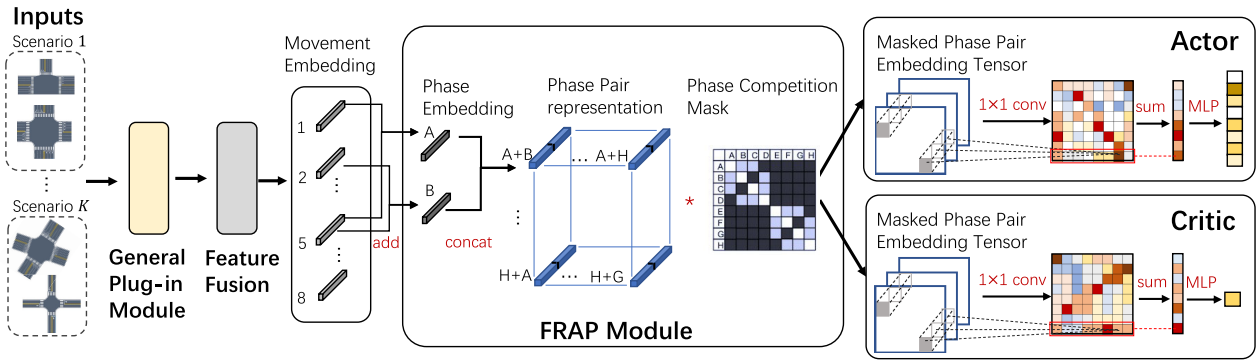


Fig. 3. Network design in our framework.

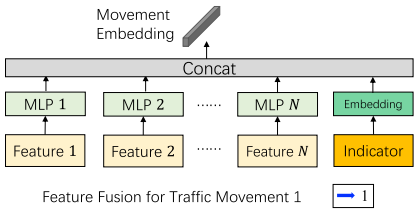


Fig. 4. A traffic movement embedding.

easily. (3) Actor and critic networks are adopted, which can be trained by A3C; thus, each progress takes one scenario to interact with the model and returns the updated gradient.

C. Detailed RL Settings

1) State: To respect an actual situation, we measure each entering lane demand within 50m of the intersection [38]. We choose five features from the observations as states: queue length, current phase, occupancy, flow, and the number of cars that are stopping [1].

2) Action: The duration of each phase is fixed in our setting, so the action is to decide which phase from the eight options (A-H) to choose for the next based on the current states.

3) Reward: The reward is a weighted average of four reward components: queue length (L_i), wait time (T_i^w), delay time (T_i^d), and pressure (P_i), with detailed definitions and calculations in Appendix A. The reward R_t for the step t: $R_t = \sum_{c \in C} w_c c_i$, where w_c re-scales the value of each component c, C is the set of components $C = 4$, and c_i is the reward component c in intersection i at step t. The smaller the four reward components are, the better the traffic condition is. To maximize the total reward, the weights are set to negative. Suggested IntelliLight [7], all the sub-rewards, after re-weighting, should be in a similar scale, given queue length, wait time, and delay time are equally important for an optimally controlled intersection. Thus, in our experimental, we decided the weights according to parameter tuning so that the four reward components are in a similar scale of $[-1, 0]$, and we set $w_{queue\ length} = -1e^{-3}$, $w_{wait\ time} = -1e^{-3}$, $w_{delay\ time} = -1e^{-5}$, $w_{pressure} = -5e^{-3}$.

V. Experiment

A. SUMO Scenarios

We choose the simulation of urban mobility (SUMO) [39] as the simulator. Seven different scenarios are employed for evaluation, and each one includes multiple intersections within a region. Four of them are publicly available [40]: *Grid* 4×4, *Arterial* 4×4, *Ingolstadt 21* [41],[1] and *Cologne 8* [42].[2] As another contribution for RL-based TSC, we build three scenarios, i.e., *Grid* 5×5, *Fenglin*, and *Nanshan*. We build *Fenglin* via the SUMO *netedit* module based on a real-world scenario and its traffic flow data is simulated based on the actual flow, more details are written in Appendix V-A. We also built *Nanshan* with 28 traffic lights via *OpenStreetMap* based on a real district in Shenzhen, China. During training, we used all seven scenarios, and our model does not need manual labels. During testing, we evaluated the performance of the models using four public scenarios and the *Fenglin* scenario. The former allows for a fair comparison between each model, while the latter enables evaluation of the model performance based on real-world traffic flow and road network. The details about the five open source scenarios are introduced in Appendix B. Some key indexes of 7 scenarios are shown in Table I, indicating the number of different intersections (Int.).

TABLE I
Statistics of the Scenarios

Scenario	# Int.	# 2-way Int.	# 3-way Int.	# 4-way Int.
Grid 4×4	16	0	0	16
Arterial 4×4	16	0	0	16
Ingolstadt 21	21	0	17	4
Cologne 8	8	1	3	4
Grid 5×5	25	0	0	25
Fenglin	7	0	2	5
Nanshan	28	1	6	21

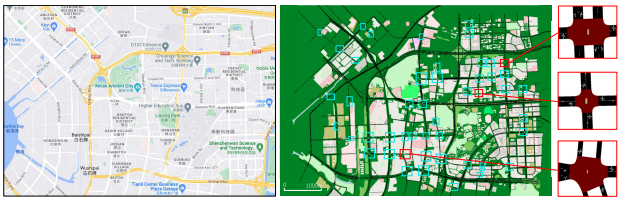
Two Real-world Scenarios: To build more realistic scenarios, we manually construct the *Nanshan* scenario based on the Nanshan district in Shenzhen, China, and *Fenglin* scenario based on the Fenglin corridor in Shaoxin, China.

For the *Nanshan* scenario, we generate the road network based on OpenStreetMap, and we set the flow pattern and the initial phase setting as the SUMO default.

For *Fenglin* scenario, it is even more elaborately designed such that it is as real as possible. As illustrated in Fig. 6.(a), It is a corridor network in Shaoxing, China, and seven intersections are signal-controlled. There are four main steps when constructing this scenario:

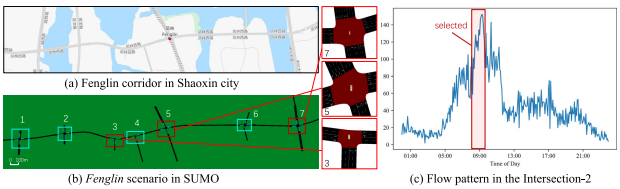
[1]https://sumo.dlr.de/docs/Data/Scenarios/TAPASCologne.html
[2]https://github.com/silaslobo/InTAS

Authorized licensed use limited to: Erasmus Universiteit Rotterdam. Downloaded on February 25,2025 at 07:07:54 UTC from IEEE Xplore. Restrictions apply.



(a) Nanshan District in Shenzhen (b) *Nanshan* scenario in SUMO

Fig. 5. (a) Nanshan district; (b) *Nanshan* scenario in SUMO, with the signaled intersections highlighted and three intersections zoomed for demonstration.



(a) Fenglin corridor in Shaoxin city (b) *Fenglin* scenario in SUMO (c) Flow pattern in the Intersection-2

Fig. 6. (a) Fenglin corridor; (b) *Fenglin* scenario in SUMO, with the signaled intersections highlighted and three intersections zoomed for demonstration; and (c) Flow pattern at the intersection-2 during the morning peak hour around 9 AM.

- Constructing the road network: We built the road network via the SUMO *netedit*[3] module based on the Fenglin West Road in Shaoxin;
- Setting the movements for each entering approach: For example, the north entering approach has two through lanes, two left lanes, and one through-right lane;
- Setting the traffic signal initial phase: we use the default setting;
- Setting the traffic flow: For each intersection, live cameras are deployed to count the passing vehicles and calculate the average traffic flow, as shown in Fig. 6.(c). (1) We select the peak hour in the red box and inject the flow from the boundary intersection, e.g., intersection 1, into the network; (2) The flow split ratio also needs to be defined. We calculate it based on the real data for each entering approach. For example, the ratio of through:right:left for intersection 5 is 5:3:2; (3) Lastly, we place 50-meter Lane-Area (E2) detectors[4] in each entering lane to get the observations such as queue length and occupancy. We place Multi-Entry-Exit Detectors (E3)[5] between the entry and exit to get the delay time and wait time.

After the aforementioned designs, the *Fenglin* scenario reflects reality very well.

B. Baseline Methods

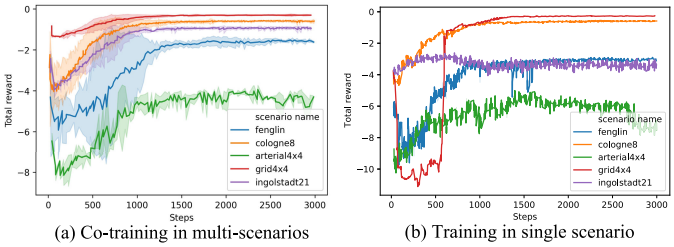
Several representative peers are selected, including both traditional methods and RL-based methods.

- **Fixed-timed control (FTC)** [18]: A fixed cycle time and phase order are set manually for each traffic light according to expert knowledge.
- **MaxPressure** [21], [22]: As a popular choice in traditional TSC, it sets the phase with the max pressure between phases as green according to the queue length.

[3]https://sumo.dlr.de/docs/Netedit/index.html

[4]https://sumo.dlr.de/docs/Simulation/Output/Lanearea_Detectors_%28E2%29.html

[5]https://sumo.dlr.de/docs/Simulation/Output/Multi-Entry-Exit_Detectors_%28E3%29.html



(a) Co-training in multi-scenarios (b) Training in single scenario

Fig. 7. Learning curves in all scenarios.

- **MPLight** [13]: It is an extension work based on FRAP. It improves FRAP by utilizing pressure as both the state and reward for a DQN agent.
- **MetaLight** [15]: It combines Meta RL with FRAP. However, it needs heavy labeling work, and the training scenarios are quite similar.
- **GESA-single**: We also test our GESA's performance in a single scenario training setting.
- **GESA** (The proposed model): It is the only method that can be co-trained in multiple scenarios.

C. Evaluation on Multiple Scenarios Co-Training

1) GESA Outperforms Baselines: The GESA is co-trained with all seven scenarios. The numerical results of all methods in five scenarios are summarized in Table II, with the best results in boldface and the second-best underlined. Compared to the baselines, our GESA achieves the highest total reward over the four scenarios and the second-highest in the *Arterial* 4×4. On average, we achieve $+$**13.27%** improvement compared to the best baselines and $+$**14.38%** than MetaLight.

In the *Arterial* 4×4, we achieve the best sub-rewards in queue length and delay time. This may be because for some scenarios, there are conflicts between the sub-rewards, making it difficult for GESA to promote them in all the scenarios simultaneously. In contrast, MetaLight takes the advantage of only training two scenarios each time. However, this comes with a curse that MetaLight's training is quite unstable due to the heterogeneity of our scenarios, as shown in Fig. 16 in Appx. E, thus, it cannot beat GESA in all other scenarios.

In the *Grid* 4×4, the GESA-single model obtains higher rewards than the full GESA since this scenario is straightforward with low traffic volume. However, in more complex scenarios, GESA-single performs worse since it tends to get stuck in a local optimum.

2) Co-Training can Converge Faster and More Stable: The learning curves in each scenario are illustrated in Fig. 7(a). We observe that training on *Grid* 4×4 is the most stable with the highest reward due to its simple topology. The training variance on *Nanshan* and *Fenglin* scenarios are the largest since their settings are realistic and complex. Compared with GESA trained in the single scenario in Fig. 7(b), co-training can also converge faster and more stable.

D. Dealing With a New Scenario

GESA's Transfer is Still Better: To study the transferability of the model, we perform the leave-one-out operation, namely,

Authorized licensed use limited to: Erasmus Universiteit Rotterdam. Downloaded on February 25,2025 at 07:07:54 UTC from IEEE Xplore. Restrictions apply.

TABLE II
LARGE SCALE CO-TRAINING PERFORMANCE

Model	Rewards	*Grid 4×4*	*Arterial*	*Ingolstadt 21*	*Cologne 8*	*Fenglin*
FTC (single-scenario)	Total	-0.6376	-8.1955	-1.9223	-1.9565	-4.2271
	Delay time	662.6	6632.4	3066.5	2094.3	4250.4
	Wait time	280.6	1183.3	1129.8	1193.3	1852.9
	Queue length	205.8	2274.7	523.9	517.0	1512.5
	Pressure	329.0	1860.5	647.5	606.5	2901.0
MaxPressure (single-scenario)	Total	-0.4024	-7.9282	-1.1089	-0.6583	-1.7214
	Delay time	545.7	6032.1	2958.8	1353.8	2849.2
	Wait time	173.7	1150.8	742.0	397.8	808.6
	Queue length	116.4	1857.2	221.3	171.3	548.2
	Pressure	248.0	2264.0	302.5	199.5	1095.5
MPLight (single-scenario)	Total	-0.4385	-4.0535	-1.1628	-0.7669	-1.8667
	Delay time	561.0	4088.9	2808.4	1481.9	3057.7
	Wait time	187.9	704.4	770.5	455.6	791.6
	Queue length	129.8	1195.0	254.5	211.2	668.3
	Pressure	268.0	2082.0	329.5	224.5	1206.5
MetaLight (multi-scenario)	Total	-0.3745	**-3.7568**	-0.8130	-0.5517	-1.7336
	Delay time	537.5	3725.1	2937.4	1298.2	2775.2
	Wait time	163.1	544.7	568.7	330.0	818.6
	Queue length	104.8	1027.0	128.8	141.2	549.2
	Pressure	235.5	**1549.5**	225.0	172.0	1108.5
GESA-single (single-scenario)	Total	**-0.2643**	-6.5587	-0.7576	-0.5503	-2.8688
	Delay time	**479.0**	4158.8	2221.4	1292.6	3181.8
	Wait time	**111.8**	**326.1**	542.0	329.9	**641.9**
	Queue length	**69.4**	1285.1	118.3	140.2	954.6
	Pressure	**168.5**	2135.0	213.5	167.0	2225.0
GESA (multi-scenario co-train)	Total	-0.2761	-3.8856	**-0.7401**	**-0.4834**	**-1.5623**
	Delay time	487.9	**3552.6**	**2200.6**	**1239.1**	**2672.2**
	Wait time	117.1	517.1	**532.6**	**290.3**	742.9
	Queue length	73.5	**975.0**	**115.3**	**120.0**	**486.8**
	Pressure	174.0	1608.0	**199.0**	**152.5**	**1033.5**
Improvement	Total	35.64%	-3.43%	9.85%	14.13%	10.18%

TABLE III
GENERALIZATION POWER OF GESA: GESA TRANSFER *v.s.* BASELINE'S SINGLE-SCENARIO TRAINING

Model	Rewards	Unseen Scenario for GESA				
		Grid 4×4	*Arterial 4×4*	*Ingolstadt 21*	*Cologne 8*	*Fenglin*
Best of the baselines	Total	-0.3745	-3.7568	-0.8130	-0.5517	-1.7214
GESA	Total	**-0.3085**	**-3.6940**	**-0.7706**	**-0.4720**	**-1.6964**
	Delay time	510.4	3888.4	2252.5	1238.6	2634.0
	Wait time	132.4	595.5	564.2	281.0	808.9
	Queue length	84.40	1108.2	114.4	118.3	535.6
	Pressure	189.0	1825.5	194.5	152.5	1145.0
Improvement	Total	21.39%	1.70%	5.50%	16.89%	1.47%

training the model on six scenarios and testing in the seventh scenario that is unseen. We rotate the unseen scenario among the five scenarios in turn, as shown in Table III, where each column shows the evaluated results on the taken-out (unseen) scenario. The performance from the second-best baseline is also listed. Our framework still achieves a 9.39% average improvement over the five scenarios, which demonstrates our model's higher generalizability. Moreover, we also observe that the performance of transfer could depend on the complexity of the unseen scenario: when transferring to an easy scenario, higher performance is expected: for example, when transferring to *Grid* 4×4 (simple road network structure, low-density traffic flow), we observe 21.39% performance gain; on the contrary, when transferring to some complex scenarios, such as *Arterial* 4×4 (quite intense traffic flow) and *Fenglin* (corridor with intense traffic flow in a bottleneck intersection), we observe relative lower performance gain, with 1.7% and 1.47% respectively.

E. Ablation Studies

We also conducted several ablation studies, with the primary ablation to show the effect of co-training with a different number of scenarios (Sec. V-E.1): it shows a steady increase of performance with more co-trained scenarios; Moreover, Sec. V-E.1.b further explains why co-training in multiple scenarios outperforms single-scenario training due to the potentials of getting stuck in a local optimum.

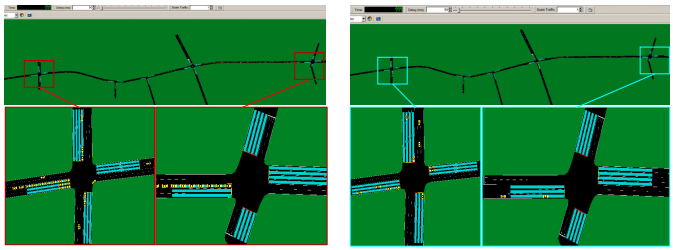
Besides, further ablation studies are shown in the appendices: (1) Appx. E-A shows the necessity of training simultaneously; (2) Ablation on rewards (Appx. E-B) shows four rewards being trained together is necessary to achieve better performance than single-reward versions.

1) Ablation Study on the Number of Co-Trained Scenarios: This subsection will examine the effect of co-training different amount of scenarios on the model performance in Sec. V-E.1.a, as well as explain in detail why multi-scenario co-training is better than single-scenario training in Sec. V-E.1.b.

Authorized licensed use limited to: Erasmus Universiteit Rotterdam. Downloaded on February 25,2025 at 07:07:54 UTC from IEEE Xplore. Restrictions apply.

TABLE IV
RESULTS OF JOINT TRAINING WITH DIFFERENT NUMBERS OF SCENARIOS BASED ON *Arterial* 4 × 4 SCENARIO'S REWARD

# scenarios	Chosen scenarios	Reward	Impr
1	***Arterial 4×4***	-6.5587	-
3	***Arterial 4×4*** + *Cologne 8*, *Nanshan*	-6.3182	+ 3.67%
5	***Arterial 4×4*** + *Cologne 8*, *Nanshan* + *Grid 4×4*, *Ingolstadt 21*	-4.3198	+ 34.14%
7	***Arterial 4×4*** + *Cologne 8*, *Nanshan* + *Grid 4×4*, *Ingolstadt 21* + *Grid5x5*, *Fenglin*	-3.8856	+ 40.76%



(a) GESA-single in *Fenglin:* More cars get stuck on the turning-left lane

(b) GESA (co-trained) in *Fenglin:* Cars on different lanes are balanced

Fig. 8. Compare single-training and co-training visualization result in *Fenglin* scenario.

a) Performance increases with more scenarios: We use *Arterial* 4 × 4 as the anchor to verify the impact of the number of co-trained scenarios because there is a big gap in its results in single-scenario v.s. multi-scenario. We increase the co-trained scenarios with two stride lengths, i.e., single, three, five, and seven, and then perform the test, respectively. The result is shown in Table IV. It can be observed that the reward increases monotonically as the number of co-training scenarios rises. This means that adding more scenarios can facilitate the model performance, and the proposed general scenario-agnostic method works very well. This may be because the different scenarios may share similar latent spatiotemporal correlation, and co-training achieves the effect of "1 + 1 > 2" by leveraging this spatiotemporal correlation.

b) Why multiple scenarios outperforms single scenario: As demonstrated in Table II and Table IV, the multi-scenario co-training is much better than the single-scenario in most cases. Through visualization, we find that single-scenario training is prone to falling into local optima than large-scale co-training.

A toy example is *Fenglin*. In Figure 8, we plot the traffic status in a particular simulation step in this scenario. One can find that the traffic controlled by the co-training model is less congested than the single-training model (the model only trained by *Fenglin* scenario) at the same time step. An obvious issue of the single-training one is that its left-turn traffic movement has a low priority of release, resulting in more queue length and pressure in the left lanes.

This phenomenon is interpretable: In terms of learning form, intersections can only obtain their own information, so they are independent learners, but their internal traffic flow is related to the same single scenario. The *Fenglin* road network is a corridor that has an East-West main road and multiple North-South branch roads. Vehicles spend much more time on the main road than on branch roads (delay and wait time, etc., are also mainly accumulated along the go-straight direction); therefore, it is easier for the agent to learn that the priority of letting go straight is higher than turning left for all intersections. But, once the agent is over-fitted in taking "go straight" action, it is difficult to break out. Because there is heavy traffic flow in the through lane at most intersections, they will get a much smaller reward for exploring turning left for a long time and even larger rewards for other intersections at the same time. This situation will not be encountered in the multi-scenario co-training model, since it handles diverse scenarios with various road networks and traffic flows, and an agent with better generalization can be learned.

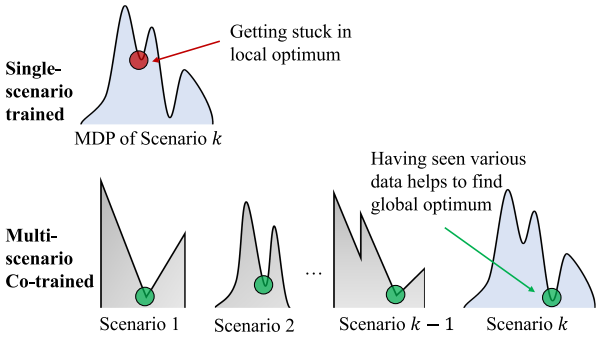
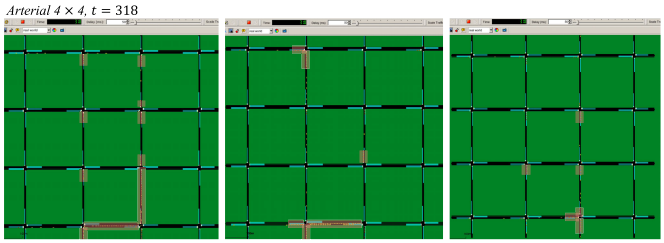


Fig. 9. Advantages of co-trained agent over single-trained agent.

This can also be explained from the perspective of the Markov Decision Process (MDP): each scenario can be viewed as a high-dimensional distribution, and the agent's interaction with the environment, the MDP process, i.e., $(s_1, a_1, r_1, s_2, a_2, r_2, \cdots)$, is exploring and learning the distribution of the scenario. As demonstrated in Fig. 9, an agent that has seen various distributions can easily realize that its current policy could be only sub-optimal. Similar to our agent, since it has seen other non-corridor scenarios other than *Fenglin*, it learns how to balance different directions (e.g., turn-left, go-straight) in non-corridor scenarios instead of simply prioritizing releasing go-straight directions. To conclude, the distinctive nature of a singular scenario's road network structure and traffic flow readily leads the model to converge towards local optima. Conversely, multiple scenarios co-training yields a broader diversity of samples by incorporating unrelated samples from various scenarios. Thus, it diminishes the likelihood of falling into local optima compared to training on a single scenario.

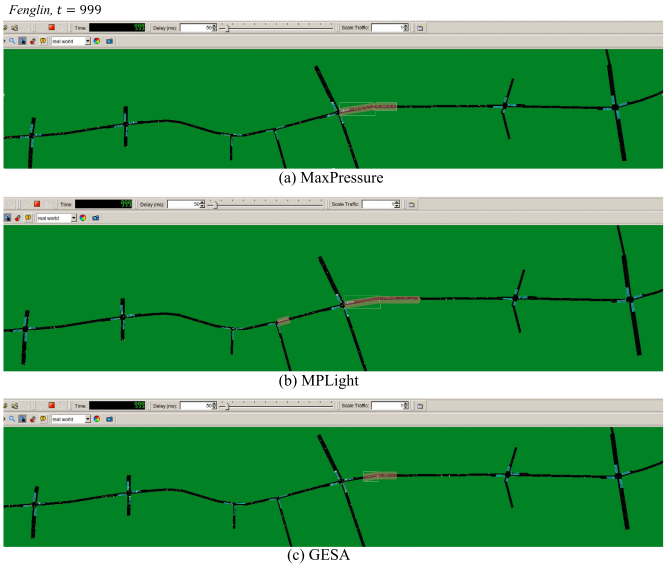
This reason can also explain why the improvement of multi-scenario performance is trivial compared to the one in *Grid* 4 × 4, the single scenario. The road network of *Grid* 4 × 4 scenario is symmetrical, and the distribution of traffic flow is smooth and relatively uniform. While in *Arterial* 4 × 4, even if its road network is very symmetrical, its traffic flow distribution is very uneven, the top and tail row intersections have heavy traffic flow, and two middle row intersections have no traffic flow in East-West lanes.

Authorized licensed use limited to: Erasmus Universiteit Rotterdam. Downloaded on February 25,2025 at 07:07:54 UTC from IEEE Xplore. Restrictions apply.



(a) FTC (b) GESA-Single (c) GESA

Fig. 10. Comparison in *Arterial* 4 × 4 with the methods (a) FTC, (b) GESA-Single, and (c) GESA. (Red highlights congestion).



(a) MaxPressure

(b) MPLight

(c) GESA

Fig. 11. Comparison in *Fenglin* with the methods (a) MaxPressure, (b) MPLight, and (c) GESA.

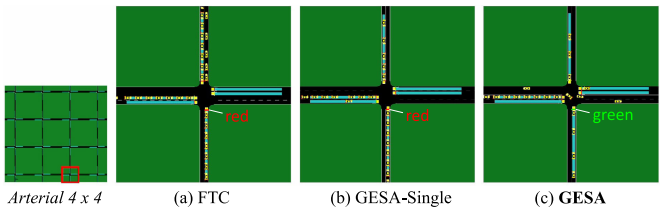
F. Demo Visualizations

In this section, we will demonstrate the effectiveness of our model for real-time traffic signal control across three comparing models on three selected scenarios. The real-time visualization is available in the link https://github.com/AnonymousIDforSubmission/GESA, and we also offer some snapshots for comparison on each scenario at the same timestamp.

1) Visualization for Scenarios: The traffic status during simulating for *Arterial* 4 × 4 and *Fenglin* are illustrated in Figures 10 and 11, respectively. The congestion queue is highlighted in red. However, we would like to emphasize that in the demo, the human eyes could only visually tell the difference in queue length. Yet the whole traffic situation is evaluated by four rewards from different perspectives. And the metrics such as delay and wait times are not easily perceived visually.

Fig. 10 shows the 318-th simulation step in the *Arterial* 4×4 scenario. Due to the high traffic volume, longer queues are formed in several intersections when the traffic is controlled by the FTC method (Fig. 10(a)). The GESA-Single model (Fig. 10(b)) can alleviate the condition to some extent. In contrast, our GESA model (Fig. 10(c)) shortens the queue length significantly, which demonstrates its superiority intuitively.

Similarly, we also list the snapshots in Fig. 11 in *Fenglin* scenario where the traffic flow is generated based on the real-world peak hour. The long queues in the central intersection are also formed when conducting MaxPressure (Fig. 11.(a)) and MPLight (Fig. 11(b)), while our GESA model still remains a shorter queue.



Arterial 4 x 4 (a) FTC (b) GESA-Single (c) **GESA**

Fig. 12. Comparison in one intersection of *Arterial* 4 × 4 with the methods (a) MaxPressure, (b) MPLight, and (c) GESA.

2) Visualization for One Intersection: Besides, we further zoom in to visualize some intersections. As shown in Fig. 12, we picked the intersection in the red rectangle when it experienced a high peak inflow demand in the through movement of the south approach. The methods of FTC and GESA-Single still turn red light to the through movement of the south approach, resulting in a long queue there. However, the proposed GESA managed to turn the green light timely.

G. Discussion on Scalability

Though the primary objective of GESA is to address the challenges related to transferability and generalization across various scenarios rather than focusing on solving large-scale applications, our approach is still easily applicable across a city due to its decentralized nature, making decisions solely based on observations of a single intersection. Thirdly, in reality, when controlling the traffic signals of a whole city, it is a common practice to partition the city into multiple subareas according to the intersections' similarities in structure and traffic conditions [43], with a subarea usually 4-7 intersections, spatial spanning within 1 kilometer. This subarea control strategy is proven to be more effective and flexible when the real traffic conditions can differ from different districts, and when traffic incidents happen, the impact can be better managed within a subarea. Thus, when applying GESA to large-scale cities, the same subarea control strategy can be adopted.

VI. Conclusion

In this paper, we propose GESA, a general RL-based framework for traffic signal control (TSC), which can obtain a general model through large-scale training with multiple scenarios. We first present a general plug-in module to eschew manual annotation in various intersections. To keep the model input and output consistent in different scenarios, we elaborately designed the unified state space and action space. Based on these unification operations, we modify the original FRAP to a policy gradient-based framework, which facilitates the model coverage and is compatible with different intersections.

We conducted prudent experiments which show that our framework outperforms the baselines. Moreover, there are valuable insights that we believe contribute significantly to the RL-based TSC community: (1) co-training in multiple scenarios might improve the agent's decision, and moreover,

Authorized licensed use limited to: Erasmus Universiteit Rotterdam. Downloaded on February 25,2025 at 07:07:54 UTC from IEEE Xplore. Restrictions apply.

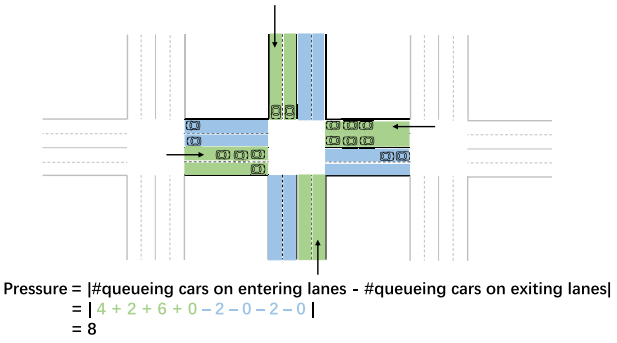


Fig. 13. The illustration of pressure.

the more scenarios co-trained together, the agents might make better decisions for TSC. The improvements may be because the co-trained agents observe more diverse situations, being less likely to get stuck in a local optimum; (2) the co-trained model, when transferring to a new unseen scenario, could even perform better than the model trained in that single scenario.

In the future, we will further narrow the gap between simulation and real-world implementation. Thanks to the model generalization is improved significantly, in the future, we plan to implement the GESA model in real cities. We will also consider the intersections with more than four entering approaches.

Appendix A
Reward Specifications

As shown in Section IV-C, the weighted combination of four components is adopted as the reward function. Their definitions are listed as follows.

Delay Time T_i^d: The delay time for the intersection i is the average of all the lanes' delay time. The delay time for a lane is the sum of all vehicles' delay time on the lane. A vehicle delay time d_v is calculated by:

$$d_v = \Delta t \left(1 - \frac{v_v(t)}{v_{max}}\right), \tag{6}$$

where Δt is interval time, $v_v(t)$ is the average speed over the interval time, and v_{max} is the max speed of vehicle v. Its unit is second (s).

Wait Time T_i^w: The wait time for the intersection i is the average of all the lanes it contains, whereas the wait time for a particular lane is the sum of all-vehicles stop time on it. The unit is second (s).

Queue Length L_i: The queue length is defined as the average of all-lanes vehicle queue length in an intersection, which is in meter (m).

Pressure P_i: As it is first introduced in [13] and as shown in Fig. 13, the pressure of the intersection i is the difference between the queuing vehicles numbers on entering and exiting approaches. Intuitively, a higher pressure indicates a greater imbalance between the entering and exiting lanes.

Appendix B
The Details of Open-Source Scenarios

The details for *Grid* 4 × 4 [13], *Grid* 5 × 5, and *Arterial* 4 × 4 [44] are given in Figure 14. The *Grid* 5 × 5 is designed by ourselves, following the same setting of *Grid* 4 × 4.

The scenarios *Ingolstadt 21* [41][6] and *Cologne 8* [42][7] describe the one-day traffic within the city of Ingolstadt and Cologne in Germany, respectively, as shown in Figure. 15.

Appendix C
Training Parameters Setting

The total simulation steps of each scenario are 3600*s*, and the agent interacts with the simulation environment every 15 seconds. During the testing stage, we repeatedly evaluate the method five times under different random seeds, and the average is regarded as the final result. For GESA, the learning rate is set as 10^{-4}, the batch size is 128, and the rollout length is set to 128. Generalized advantage estimation[8] is adopted as the value estimation method. After each rollout and return of the gradient, the shared model updates parameters according to the newest parameter in each process of A3C [36].

Appendix D
FTC Configuration

To ensure a fair comparison, when different models are interacting with the same scenario, all the scenario-related configurations are the same for all different methods, such as the same long simulation time and traffic flow pattern.

When different methods are giving their traffic control policies, it is worth mentioning that all the methods, except for FTC, are based on phase selection: to select the next phase from A to H, and the execution duration of each phase is fixed as 10 seconds, followed by a 5-second yellow light time. Since there is no fixed order of different phases, these methods are also known as "acyclic methods".

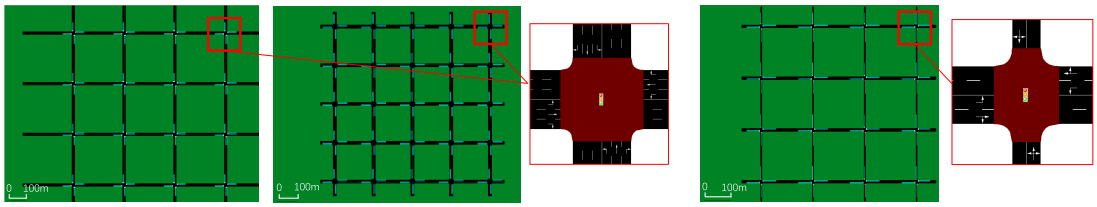
FTC instead is a cyclic method where the phase order is fixed; for example, E-H-A-D is a commonly chosen phase order in real cities. For cyclic methods, only the duration for each phase needs to be specified. Fixed Time Control, as it is named, has a fixed phase duration for a given intersection regardless of the dynamic traffic, and the optimal phase duration is usually calculated by analytic solutions, such as Webster's method [5]: $C = \frac{1.5 \cdot L + 5}{1 - Y}$ where L is the loss time, and Y is the sum of the critical flow ratio.

To explain FTC's action parameter settings more clearly, we list the actions of the main intersections in each evaluation scenario and skip the 3s or 5s yellow light time:

- *Fenglin*: This scenario is based on the real city network and traffic-flow data, the default actions of all 7 intersections are also based on the real situation. All action parameters are shown in Table V.
- The rest of the scenarios presented in the main paper are open-source scenarios. The FTC parameters we use in these scenarios are the scenarios default. In *Arterial* 4 × 4 and *Grid* 4 × 4, all intersections within each scenario use the same parameters, specifically: *Arterial* 4 × 4: E-10s, D-10s, A-10s, B-10s, and C-10s. *Grid* 4 × 4: E-10s, H-10s, F-10s, G-10s, A-10s, D-10s, C-10s, and B-10s.

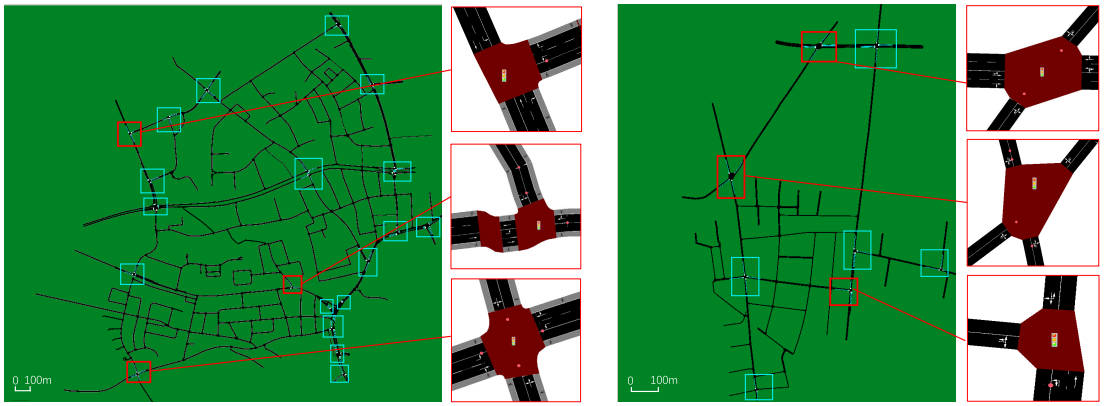
[6]https://sumo.dlr.de/docs/Data/Scenarios/TAPASCologne.html
[7]https://github.com/silaslobo/InTAS
[8]https://arxiv.org/abs/1506.02438 and https://github.com/topics/generalized-advantage-estimation

Authorized licensed use limited to: Erasmus Universiteit Rotterdam. Downloaded on February 25,2025 at 07:07:54 UTC from IEEE Xplore. Restrictions apply.



(a) *Grid 4×4* (b) *Grid 5×5* (c) *Arterial 4×4*

Fig. 14. The scenarios of (a) *Grid* 4×4, (b) *Grid* 5×5, with all the intersections signaled and each entering approach having three lanes with movements of left, through, and right, respectively; (c) *Arterial* 4×4, with all the intersections signaled, E & W entering approaches having two lanes with movements of left and right-through, as N & W entering approaches having one lane with the movement of left-through-right. (The blue strips indicate the locations of 50-meter detectors.)



(a) *Ingolstadt 21* (b) *Cologne 8*

Fig. 15. The scenarios of (a) *Ingolstadt 21*, (b) *Cologne 8*, with the signaled intersections highlighted and three intersections zoomed for demonstration.

In *Cologne8*, there are two main types: (1) A-33s, D-6s, E-33s, and H-6s. (2) E-38s, H-6s, and D-37s. In *Ingolstadt 21*, there are three main types: (1) A-38s, D-6s, and E-37s. (2) A-35s, C-6s, and H-34s. (3) A-29s, D-6s, E-29s, and H-6s.

APPENDIX E
ABLATION STUDIES

In this section, detailed ablation studies will be conducted to further demonstrate the necessity of our model design. Section E-A will compare the effect of training A3C in a sequential manner. Section E-B studies the effects of each reward. Section V-E.1 studies how introducing different amounts of scenarios for co-training will affect performance.

A. Ablation Study on the Model Components

(1) Co-training together is better than co-training in a sequential As shown in Fig.16(a), we display the total reward of MetaLight in each checkpoint during evaluation. Since the Meta-Learner in MetaLight is trained with only two scenarios observed each time, and our scenarios are quite heterogeneous to each other, so the evaluating results are quite unstable, with the total rewards fluctuating largely. Our GESA instead is improving total reward more steadily, as shown in Fig.7(a) and Fig.16(c).

To validate the importance of the A3C module, we train the GESA model in a sequential manner (i.e., feeding the scenarios one by one to train the model) rather than the original parallel manner (i.e., training the model with all the scenarios simultaneously). This variant is termed as GESA-sequential-training, and its training curves are shown in Fig.16(b). The rewards of all the models are small and without an upward trend, which means that they do not converge toward favorable outcomes and the sequential manner is not applicable to the training. Compared to it, the parallel manner (implemented by A3C) can significantly improve the rewards (Fig.16(c)), which contributes to the training.

B. Ablation Study on Rewards

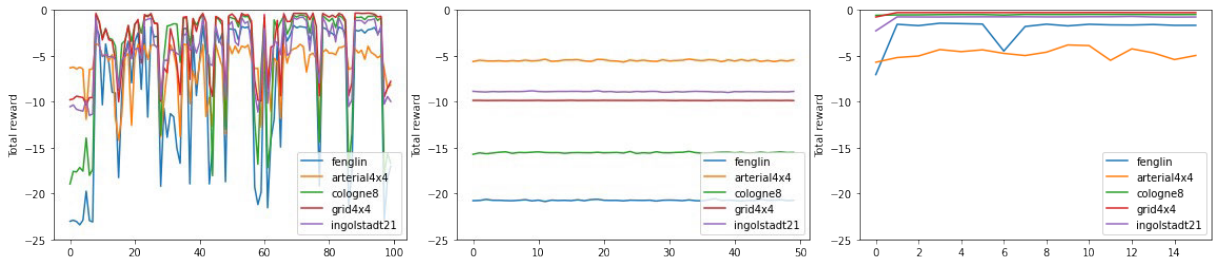
Since the designed reward functions contain four components (that is, queue length, wait time, delay time, and pressure), we perform the component ablation experiments to provide insights into their impact on the model performance specifically.

We conduct a single-component v.s. all-component reward ablation experiment in the co-training setting, the results are shown in Table VI. Grey values represent that the corresponding components are excluded from the reward function during training but measured in the evaluation. The best performance is in boldface and the near-best one is underlined. It can be seen that even though the GESA trained with a single component may outperform under the particular metric, its performance on the remaining metrics is unremarkable. For example, in the scenario *Ingolstadt 21*, $\text{GESA}_{\text{wait time}}$ achieves the best on the wait time metric but degrades about 150.5%, 1522.4%, 1577.1% on the delay time, queue length, and pressure compared to the canonical GESA, respectively. The

Authorized licensed use limited to: Erasmus Universiteit Rotterdam. Downloaded on February 25,2025 at 07:07:54 UTC from IEEE Xplore. Restrictions apply.

TABLE V
THE FTC PARAMETERS OF EACH INTERSECTION OF *Fenglin* SCENARIO

Type	Phase1	Time1(s)	Phase2	Time2(s)	Phase3	Time3(s)	Phase4	Time4(s)	Number	way Int.
1	E	45	H	28	A	54	D	31	1	4-way
2	E	44	H	28	A	69	D	39	1	4-way
3	A	42	H	29	-	-	-	-	1	3-way
4	A	39	H	27	-	-	-	-	1	3-way
5	E	38	H	25	A	54	D	27	2	4-way
6	E	31	H	18	A	28	D	18	1	4-way



(a) Evaluation of MetaLight (b) Evaluation of GESA-sequential-train (c) Evaluation of GESA

Fig. 16. (a) Evaluation of MetaLight; (b) Evaluation of GESA with the Actor-Critic feeding the scenarios in a sequential manner; (c) Evaluation of GESA.

TABLE VI
SINGLE-COMPONENT VS 4-COMPONENT REWARD

Component	Rewards	*Grid 4x4*	*Arterial 4×4*	*Ingolstadt 21*	*Cologne 8*	*Fenglin*
$GESA_{\text{delay time}}$	Total	-0.3266	-4.8884	-0.8715	-0.5847	-1.5905
	Delay time	511.7	3906.4	2267.1	1266.1	2855.5
	Wait time	139.4	420.8	594.2	346.3	**563.6**
	Queue length	89.9	1137.7	154.7	154.3	550.2
	Pressure	210.0	1960.5	276.5	188.0	1395.0
$GESA_{\text{wait time}}$	Total	-10.1055	-4.8713	-7.1326	-11.1791	-16.4658
	Delay time	5642.4	2607.8	5513.5	7698.9	11995.5
	Wait time	655.2	**189.3**	**516.1**	674.6	977.4
	Queue length	2907.6	985.4	1870.6	3201.8	3332.4
	Pressure	5121.0	2178.0	3337.5	3369.5	14766.0
$GESA_{\text{queue length}}$	Total	-0.3005	-5.4331	-0.9334	-0.5207	-1.6324
	Delay time	516.0	4071.8	2258.4	1291.0	2636.7
	Wait time	127.4	515.6	624.4	315.8	749.2
	Queue length	82.60	1175.6	182.5	126.3	518.7
	Pressure	186.5	1850.5	281.0	167.0	1149.0
$GESA_{\text{pressure}}$	Total	-0.3261	-6.3235	-1.2017	-0.5417	-1.5831
	Delay time	512.4	4306.6	2637.2	1334.8	**2598.8**
	Wait time	138.4	607.1	792.2	326.6	721.5
	Queue length	91.0	1386.4	228.1	132.7	517.2
	Pressure	204.5	1900.0	354.5	172.5	1107.5
GESA	Total	**-0.2761**	**-3.8856**	**-0.7401**	**-0.4834**	**-1.5623**
	Delay time	**487.9**	**3552.6**	**2200.6**	**1239.1**	2672.2
	Wait time	**117.1**	517.1	532.6	**290.3**	742.9
	Queue length	**73.5**	**975.0**	**115.3**	**120.0**	**486.8**
	Pressure	**174.0**	**1608.0**	**199.0**	**152.5**	**1033.5**

higher delay time and lower wait time indicate that the vehicles move slower even though stop infrequently, and there is a higher potential for jams accordingly.

Overall, given the intrinsic correlations in the four components, it is easy to fall into the sub-optimum when only optimizing one specific component. The canonical GESA with a four-component reward achieves promising results in most cases. The highest total reward is more straightforward to demonstrate its superiority.

APPENDIX F
DISCUSSION AND APPRECIATION

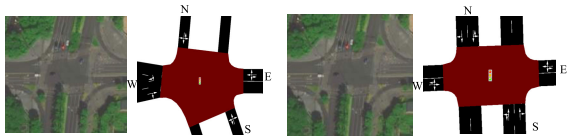
In this section, we will provide additional discussions, future work, and our appreciation.

A. Discussion About Intersections With More Than Four Approaches

By the proposed GPI module (as shown in Algorithm 1), the intersections with no more than four approaches can be handled. It should be noted that it is a demonstration and the threshold "four" is not fixed, that is, intersections with five, six, or even more approaches can also be treated following a similar pipeline. When the threshold is enlarged, one needs to (1) modify the angles interval in Algorithm 1, (2) update the conflict matrix of movements (in Fig. 2(b)) and available phases, and (3) enlarge the model output (i.e., action space).

The reasons why we choose "four" are: (1) For the intersections in real-world road networks, two-approach, three-approach, and four-approach ones occupy dominant

Authorized licensed use limited to: Erasmus Universiteit Rotterdam. Downloaded on February 25,2025 at 07:07:54 UTC from IEEE Xplore. Restrictions apply.



(a) 6-legged Intersection Example 1 in *Nanshan* (b) 6-legged Intersection Example 2 in *Nanshan*

Fig. 17. Examples of 6-legged intersections with only four entering approaches.

proportions.[9] (2) According to our practical statistics in Nanshan and Fenglin districts, there are no intersections with more than four entering approaches, so setting "four" in the proposed GPI module is quite enough to handle the realistic scenarios. Overall, the $>$ 4-approach intersections are relatively rare, and the design for such intersections in the GPI module is not worth the cost. If needed, we will consider to extend the GPI module to handle such situations specifically in our future work.

Moreover, one may be confused with the terms legs and approaches for an intersection. For example, a six-legged intersection can still have only four entering approaches, as shown in Fig. 17. The rest two legs are exiting approaches without the need for signal control.

ACKNOWLEDGMENT

The authors would like to thank the product team, which helped commercialize the model for industry applications. This work was conducted collaboratively among the authors.

Haoyuan Jiang and Ziyue Li led the model design, project management, conducting experiments, and paper writing; Zhishuai Li contributed to the first phase of the algorithm design; Lei Bai and Hangyu Mao contributed to the ideation and paper polishing; Wolfgang Ketter contributed to the paper polishing; and Rui Zhao contributed to the key ideation and project support.

REFERENCES

[1] H. Wei, G. Zheng, V. Gayah, and Z. Li, "Recent advances in reinforcement learning for traffic signal control: A survey of models and evaluation," *ACM SIGKDD Explor. Newslett.*, vol. 22, no. 2, pp. 12–18, Jan. 2021.

[2] W. Ketter, K. Schroer, and K. Valogianni, "Information systems research for smart sustainable mobility: A framework and call for action," *Inf. Syst. Res.*, vol. 34, no. 3, pp. 1045–1065, Sep. 2023.

[3] P. Hunt, D. Robertson, R. Bretherton, and M. C. Royle, "The SCOOT online traffic signal optimisation technique," *Traffic Eng. Control*, vol. 23, no. 4, pp. 190–192, 1982.

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

[5] P. Koonce and L. Rodegerdts, *Traffic Signal Timing Manual*. Washington, DC, USA: Federal Highway Administration, 2008.

[6] E. Van der Pol and F. A. Oliehoek, "Coordinated deep reinforcement learners for traffic light control," in *Proc. Learn. Inference Control Multi-Agent Syst.*, vol. 1, 2016, pp. 21–38.

[7] H. Wei, G. Zheng, H. Yao, and Z. Li, "IntelliLight: A reinforcement learning approach for intelligent traffic light control," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Jul. 2018, pp. 2496–2505.

[8] H. Wei et al., "PressLight: Learning max pressure control to coordinate traffic signals in arterial network," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, vol. 1, 2019, pp. 1290–1298.

[9] K.-L.-A. Yau, J. Qadir, H. L. Khoo, M. H. Ling, and P. Komisarczuk, "A survey on reinforcement learning models and algorithms for traffic signal control," *ACM Comput. Surveys*, vol. 50, no. 3, pp. 1–38, May 2018.

[9] https://en.wikipedia.org/wiki/Intersection_(road)

[10] S. El-Tantawy, B. Abdulhai, and H. Abdelgawad, "Multiagent reinforcement learning for integrated network of adaptive traffic signal controllers (MARLIN-ATSC): Methodology and large-scale application on downtown Toronto," *IEEE Trans. Intell. Transp. Syst.*, vol. 14, no. 3, pp. 1140–1150, Sep. 2013.

[11] G. Zheng et al., "Learning phase competition for traffic signal control," in *Proc. 28th ACM Int. Conf. Inf. Knowl. Manage.*, Nov. 2019, pp. 1963–1972.

[12] Z. Zhang, J. Yang, and H. Zha, "Integrating independent and centralized multi-agent reinforcement learning for traffic signal network optimization," in *Proc. 19th Int. Conf. Auto. Agents MultiAgent Syst.*, 2020, pp. 2083–2085.

[13] C. Chen et al., "Toward a thousand lights: Decentralized deep reinforcement learning for large-scale traffic signal control," in *Proc. AAAI Conf. Artif. Intell.*, vol. 34, 2020, pp. 3414–3421.

[14] L. Zhu, P. Peng, Z. Lu, and Y. Tian, "MTLight: Efficient multi-task reinforcement learning for traffic signal control," in *Proc. ICLR 2022 Workshop Gamification Multiagent Solutions*, 2022, pp. 1–16.

[15] X. Zang, H. Yao, G. Zheng, N. Xu, K. Xu, and Z. Li, "Metalight: Value-based meta-reinforcement learning for traffic signal control," in *Proc. AAAI Conf. Artif. Intell.*, vol. 34, 2020, pp. 1153–1160.

[16] A. Oroojlooy, M. Nazari, D. Hajinezhad, and J. Silva, "Attendlight: Universal attention-based reinforcement learning model for traffic signal control," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, 2020, pp. 4079–4090.

[17] M. Wang, Y. Xu, X. Xiong, Y. Kan, C. Xu, and M.-O. Pun, "ADLight: A universal approach of traffic signal control with augmented data using reinforcement learning," 2022, *arXiv:2210.13378*.

[18] R. P. Roess, E. S. Prassas, and W. R. McShane, *Traffic Engineering*. Upper Saddle River, NJ, USA: Prentice-Hall, 2004.

[19] M. Fellendorf, "VISSIM: A microscopic simulation tool to evaluate actuated signal control including bus priority," in *Proc. 64th Inst. Transp. Eng. Annu. Meeting*, vol. 32, 1994, pp. 1–9.

[20] P. Mirchandani and L. Head, "A real-time traffic signal control system: Architecture, algorithms, and analysis," *Transp. Res. C, Emerg. Technol.*, vol. 9, no. 6, pp. 415–432, Dec. 2001.

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

[22] A. Kouvelas, J. Lioris, S. Fayazi, and P. Varaiya, "Maximum pressure controller for stabilizing queues in signalized arterial networks," *Transp. Res. Rec., J. Transp. Res. Board*, vol. 2421, no. 1, pp. 133–141, 2014. [Online]. Available: http://journals.sagepub.com/doi/10.3141/2421-15

[23] L. Zhang, Q. Wu, J. Shen, L. Lü, B. Du, and J. Wu, "Expression might be enough: Representing pressure and demand for reinforcement learning based traffic signal control," in *Proc. Int. Conf. Mach. Learn.*, 2022, pp. 26645–26654.

[24] M. Noaeen et al., "Reinforcement learning in urban network traffic signal control: A systematic literature review," *Expert Syst. Appl.*, vol. 199, Aug. 2022, Art. no. 116830.

[25] R. Ahadi, W. Ketter, J. Collins, and N. Daina, "Cooperative learning for smart charging of shared autonomous vehicle fleets," *Transp. Sci.*, vol. 57, no. 3, pp. 613–630, May 2023.

[26] L. A. Prashanth and S. Bhatnagar, "Reinforcement learning with average cost for adaptive control of traffic lights at intersections," in *Proc. 14th Int. IEEE Conf. Intell. Transp. Syst. (ITSC)*, Oct. 2011, pp. 1640–1645.

[27] N. Casas, "Deep deterministic policy gradient for urban traffic light control," 2017, *arXiv:1703.09035*.

[28] L.-H. Xu, X.-H. Xia, and Q. Luo, "The study of reinforcement learning for traffic self-adaptive control under multiagent Markov game environment," *Math. Problems Eng.*, vol. 2013, pp. 1–10, Jan. 2013.

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

[30] T. Nishi, K. Otaki, K. Hayakawa, and T. Yoshimura, "Traffic signal control based on reinforcement learning with graph convolutional neural nets," in *Proc. 21st Int. Conf. Intell. Transp. Syst. (ITSC)*, 2018, pp. 877–883.

[31] H. Wei et al., "CoLight: Learning network-level cooperation for traffic signal control," in *Proc. 28th ACM Int. Conf. Inf. Knowl. Manag.*, 2019, pp. 1913–1922.

Authorized licensed use limited to: Erasmus Universiteit Rotterdam. Downloaded on February 25,2025 at 07:07:54 UTC from IEEE Xplore. Restrictions apply.

[32] H. Zhang, C. Liu, W. Zhang, G. Zheng, and Y. Yu, "GeneraLight: Improving environment generalization of traffic signal control via meta reinforcement learning," in *Proc. Int. Conf. Inf. Knowl. Manage.*, 2020, pp. 1783–1792.
[33] I. Goodfellow, "Generative adversarial networks," *Commun. ACM*, vol. 63, no. 11, pp. 139–144, 2020.
[34] L. Zhu, P. Peng, Z. Lu, and Y. Tian, "MetaVIM: Meta variationally intrinsic motivated reinforcement learning for decentralized traffic signal control," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 11, pp. 1–18, Nov. 2023.
[35] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," 2017, *arXiv:1707.06347*.
[36] V. Mnih et al., "Asynchronous methods for deep reinforcement learning," in *Proc. Int. Conf. Mach. Learn.*, 2016, pp. 1928–1937.
[37] I. Osband, C. Blundell, A. Pritzel, and B. V. Roy, "Deep exploration via bootstrapped DQN," in *Proc. 30th Int. Conf. Neural Inf. Process. Syst.*, 2016, pp. 4033–4041.
[38] T. Chu, J. Wang, L. Codecà, and Z. Li, "Multi-agent deep reinforcement learning for large-scale traffic signal control," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 3, pp. 1086–1095, May 2019.
[39] M. Behrisch, L. Bieker, J. Erdmann, and D. Krajzewicz, "SUMO—Simulation of Urban MObility: An overview," in *Proc. 3rd Int. Conf. Adv. Syst. Simul.*, 2011, pp. 1–6.
[40] J. Ault and G. Sharon, "Reinforcement learning benchmarks for traffic signal control," in *Proc. 35th Conf. Neural Inf. Process. Syst. Datasets Benchmarks Track*, Jun. 2021, pp. 1–11.
[41] S. C. Lobo, S. Neumeier, E. M. G. Fernandez, and C. Facchi, "InTAS—The Ingolstadt traffic scenario for SUMO," 2020, *arXiv:2011.11995*.
[42] C. Varschen and P. Wagner, "Mikroskopische modellierung der personenverkehrsnachfrage auf basis von zeitverwendungstagebüchern," *Integrierte Mikro-Simulation von Raum-und Verkehrsentwicklung. Theorie, Konzepte, Modelle, Praxis*, vol. 81, pp. 63–69, 2006.
[43] H. Guo, J. Cheng, Q. Peng, C. Zhu, and Y. Mu, "Dynamic division of traffic control sub-area methods based on the similarity of adjacent intersections," in *Proc. 17th Int. IEEE Conf. Intell. Transp. Syst. (ITSC)*, Oct. 2014, pp. 2208–2213.
[44] J. Ma and F. Wu, "Feudal multi-agent deep reinforcement learning for traffic signal control," in *Proc. 19th Int. Conf. Auto. Agents Multiagent Syst. (AAMAS)*, 2020, pp. 816–824.



Haoyuan Jiang received the M.S. degree in software engineering from Zhejiang University, Hangzhou, China, in 2019. Currently, he is a Research Scientist with Baidu. He has been a Researcher with SenseTime Research since 2019. His research interests include deep reinforcement learning, foundation models, and autonomous driving.



Ziyue Li is currently an Assistant Professor with the Information System Department, WiSo Faculty, University of Cologne. He is also the Chief Machine Learning Scientist with EWI. His research targets high-dimensional data mining and deep learning methodologies for real-world spatiotemporal problems. His expertise's are tensor analysis, spatiotemporal data, and statistical machine learning. Those methods have been applied to various industries, mainly in smart transportation, smart manufacturing, and multimedia. His works have been awarded various best paper awards in INFORMS, IISE, and IEEE CASE.



Zhishuai Li received the Ph.D. degree from the Institute of Automation, Chinese Academy of Sciences, in 2022. He is currently a Researcher with the Smart City Group, SenseTime Research. His research interests include intelligent transport systems and spatiotemporal data mining. He has published several papers in the ITS venues, such as IEEE ITSC Conference and IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS journal.



Lei Bai received the Ph.D. degree from UNSW Sydney. He is currently a Research Scientist with Shanghai AI Laboratory. Prior to that, he was a Post-Doctoral Research Fellow with The University of Sydney, Australia. His research interests include spatial-temporal intelligence and resource-efficient machine learning. The related applications include earth system science and intelligent transportation systems. He serves as a Reviewer or a Program Committee Member for a set of prestigious conferences and journals, such as NeurIPS, ICLR, ICML, AAAI, CVPR, ICCV, IJCAI, KDD, ECCV, IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, IEEE TRANSACTIONS ON IMAGE PROCESSING, and IEEE TRANSACTIONS ON MULTIMEDIA.



Hangyu Mao is currently a Principal Researcher with the SenseTime Smart City Group (SCG). Before that, he was a Principal Researcher with Huawei Noah's Ark Lab. He focuses on bringing intelligent decision-making techniques to the real world. He is specifically interested in reinforcement learning (RL), multi-agent reinforcement learning (MARL), and large language models (LLM). He serves as the (Senior) Program Committee Member of top AI conferences, such as NeurIPS/ICML and AAAI/IJCAI.



Wolfgang Ketter is currently a Chair Professor in information systems with the Faculty of Management, Economics, and Social Sciences, University of Cologne, Cologne, Germany, where he is also a Coordinator of the Key Research Initiative Sustainable Smart Energy and Mobility. In addition, he is a Chair Professor and the Director of the Erasmus Centre for Future Energy Business, Rotterdam School of Management, Erasmus University, Rotterdam, The Netherlands. He is also a Co-Founder of Power TAC, the annual Power Trading Agent Competition that simulates the performance of retail electricity markets in a swiftly changing energy landscape.



Rui Zhao received the Ph.D. degree from The Chinese University of Hong Kong in 2015. He is currently the Research Executive with the Smart City Group, SenseTime. He is also the lead Researcher with Qingyuan Research Institute, Shanghai Jiaotong University. His research interests include computer vision and deep learning. He has more than 100 publications, with around 10,000 citations.

Authorized licensed use limited to: Erasmus Universiteit Rotterdam. Downloaded on February 25,2025 at 07:07:54 UTC from IEEE Xplore. Restrictions apply.