
Single-agent Reinforcement Learning Model for Regional Adaptive Traffic Signal Control

Qiang Li¹, Ningjing Zeng² and Lina Yu*

¹ College of Urban Transportation and Logistics, Shenzhen Technology University, Shenzhen, Guangdong 518118, China

* Corresponding Authors, email: yulina@sztu.edu.cn

ABSTRACT

Several studies have employed reinforcement learning (RL) to address the challenges of regional adaptive traffic signal control (ATSC) and achieved promising results. In this field, existing research predominantly adopts multi-agent frameworks. However, the adoption of multi-agent frameworks presents challenges for scalability. Instead, the Traffic signal control (TSC) problem necessitates a single-agent framework. TSC inherently relies on centralized management by a single control center, which can monitor traffic conditions across all roads in the study area and coordinate the control of all intersections. This work proposes a single-agent RL-based regional ATSC model compatible with probe vehicle technology. Key components of the RL design include state, action, and reward function definitions. To facilitate learning and manage congestion, both state and reward functions are defined based on queue length, with action designed to regulate queue dynamics. The queue length definition used in this study differs slightly from conventional definitions but is closely correlated with congestion states. More importantly, it allows for reliable estimation using link travel time data from probe vehicles. With probe vehicle data already covering most urban roads, this feature enhances the proposed method's potential for widespread deployment. The method was comprehensively evaluated using the SUMO simulation platform. Experimental results demonstrate that the proposed model effectively mitigates large-scale regional congestion levels via coordinated multi-intersection control.

Keywords: Traffic signal control (TSC), Reinforcement learning (RL), Probe vehicle, Queue length; Congestion management

1. INTRODUCTION

Traffic congestion significantly impacts urban quality of life, traffic safety, environmental conditions, and economic performance, making it a critical issue in numerous cities worldwide. One of the major causes of urban road congestion is queuing at intersections. Traffic Signal Control (TSC) systems play a pivotal role in managing traffic flow at signalized intersections and offer significant potential for alleviating traffic congestion and enhancing travel efficiency.

Traditionally, traffic engineers have aimed to optimize traffic signal timings through the development of optimization models based on certain assumptions about traffic dynamics (1). While these models render TSC problems more tractable, they often depart from real-world conditions. As a result, they fail to capture the complex and stochastic scenarios of urban traffic, thereby limiting their effectiveness in prompt adaptation to dynamic traffic demands (2).

With the rapid progress in artificial intelligence, a promising learning-based approach, known as reinforcement learning (RL), has arisen to overcome the limitations inherent in traditional model-based control strategies (3). Through sustained research efforts, RL has become an extensively studied technology for the development of smart TSC systems (4-5). It has shown notable suitability for controlling complex and dynamic traffic environments (1, 6, 7, 8).

When considering the control of multiple intersections, researchers widely acknowledge that due to the rapid growth of the scale of the state space and action space, it is highly challenging to achieve effective control with a single agent framework. Therefore, the focus has moved to multi-agent control systems, which have demonstrated significant scalability potential (7, 9, 10).

However, implementing multi-agent systems is only a second-best solution for tackling scalability. TSC is inherently controlled by a single central control center (agent). This center is equipped to monitor the traffic conditions of all roads within the study area and orchestrate control of all intersections. This is fundamentally different from coordinating drones or Automated Guided Vehicles (AGVs).

Currently, probe vehicle technology is a highly viable traffic monitoring technology for TSC systems (11). Firstly, probe vehicle data spans most urban roads, avoiding the need for additional equipment to acquire and gather traffic information. By utilizing vehicles from ride-hailing services and users of navigation systems, it is possible to collect real-time trajectories for 10% to 30% of the vehicles on the road network. Secondly, probe vehicle technology can assess queue length, a metric that current sensors such as induction loops find challenging to capture (12). Queue length is an established metric that accurately reflects the congestion level on urban roads. Consequently, the alignment of TSC methods with probe vehicle systems will accelerate their widespread adoption.

In this work, we propose a single-agent RL-based regional TSC model, aligned with probe vehicle technology, contrasting with prevailing multi-agent frameworks applied in existing studies. The contributions of this work are summarized as follows:

- We propose a single-agent RL model for regional ATSC. This model centrally coordinates a single agent to orchestrate signal timing across multiple intersections,

marking a departure from conventional multi-agent RL methodologies. By removing inter-agent coordination overhead and preserving a unified policy optimization process, our model achieves significant simplicity.

- In RL environment design, both the state and reward functions utilize queue length, while the action directly regulates queue length, thereby boosting policy learning efficiency. The queue length metric adopted in this study varies marginally from established definitions. This formulation is strongly correlated with congestion states and, critically, allows reliable estimation using link travel time data from probe vehicles. This characteristic enables the proposed method to achieve citywide deployment potential.
- A simulation experiment was established to validate the proposed model. To validate the superiority of our method, we conducted experiments on synthetic datasets. Results indicate that the DreamerV3 algorithm effectively facilitates our model's training, ensuring reliable learning and optimization for traffic signal control. Moreover, for urban road network traffic signal control, our method not only efficiently mitigates congestion but also substantially decreases total road network travel time.

The remainder of this paper is structured as follows. Section 2 summarizes the related works. Section 3 delineates the RL-based TSC model. Section 4 describes the experiment design. Section 5 presents the experimental results and discussion. Finally, Section 6 concludes this work.

2.RELATED WORKS

In TSC research, optimal control seeks effective traffic signal timing to optimize flow, needing real-time adjustments to reduce congestion and waiting, but faces challenges from complex layouts, fluctuating flows, and multi-objectives(9).

Historically, traffic engineers have typically tackled traffic signal optimization by developing an optimization model under certain traffic dynamics assumptions (13-16). These simplifying assumptions ignore real-world stochasticity, leaving methods slow to adapt to shifting demand (2).

RL has recently exhibited impressive performance across diverse applications, including robotics, autonomous driving, unmanned aerial vehicle control, and video games. These advancements have fueled rising interest in applying deep RL frameworks to tackle the challenging TSC problem, yielding promising results (7, 9, 17).

However, large-scale TSC remains a substantial challenge. As intersection numbers increase, the state-action space expands exponentially, impeding the extraction of relevant information and the exploration of optimal policy solutions (9).

Most contemporary studies have adopted multi-agent approaches, divided into independent multi-agent reinforcement learning (MARL) and collaborative MARL. In independent MARL, agents train independently, whereas collaborative MARL facilitates coordination through mechanisms like message passing.

Chu et al. (18) built IA2C by running separate Advantage Actor-Critic (A2C) agents, then stabilized it as MA2C via Adjacent Intersection Fingerprint and Spatial Discount Factor—neighboring policy/traffic cues plus local-reward emphasis.

Chen et al. (10) developed a decentralized RL model for city-scale TSC, and conducted experiments in Manhattan, New York, covering 2,510 traffic signals—the largest-scale experiment to date. By minimizing the "pressure" metric (19-20), RL agents optimize vehicle distribution and maximize system throughput.

The primary advantage of independent MARL lies in its scalability. However, a major limitation is that each agent has access to only its local environment, which is dynamically perturbed by other agents' actions and learning processes, leading to training instability (non-stationarity) (9).

In contrast, collaborative MARL controls multiple intersections through coordinated agents that learn optimal policies for the entire system. Communication is critical, as agents exchange information to facilitate effective coordination. Bokade et al. (21) proposed the QRC-TSC framework, which enables agents to learn communication policies that selectively exchange variable-length messages while minimizing communication overhead.

To enhance coordination and alleviate congestion, Yan et al. (22) developed a Graph Cooperation Q-learning Network (GCQN-TSC) model, incorporating self-attention mechanisms that let agents dynamically adjust their focus based on real-time traffic conditions, facilitating broader and more efficient environmental awareness.

Prior works partitioned traffic networks into static regions to enhance global coordination. However, static partitioning fails to adapt to dynamic traffic patterns. To address this, Ma and Wu (23) introduced a novel MARL method featuring adaptive network partitioning.

TSC is fundamentally single-agent; centralized control suffices. Multi-agent schemes distort the problem and add needless complexity via limited views and messages. Single-agent RL now scales to TSC's large state-action spaces, making its use imperative and true to the problem's nature.

3.METHODOLOGY

We adopt single-agent RL to address regional TSC problems. RL is defined by an agent interacting with its environment, learning optimal strategies through continuous trial-and-error interaction with the environment, eliminating the need for prior knowledge of the environment's dynamic model.

All notations are summarized in Table 1.

In each iteration, the agent observes the environment state at time step t , denoted as S_t . This state serves as a comprehensive representation of the environment, encapsulating all necessary information for the agent to take optimal actions. The agent then executes an action a_t based on S_t to interact with the environment. Based on the current state S_t and selected action a_t , the environment transitions to a new state S_{t+1} , and the agent receives the new state S_{t+1} and a reward signal r_{t+1} from the environment, which quantifies the outcome of the

action a_t . Leveraging these trial-and-error iterations and an optimal strategy learning algorithm, the agent iteratively refines its action-selection strategy to achieve specific control goals.

The iterative process of applying RL to TSC is illustrated in Figure 1.

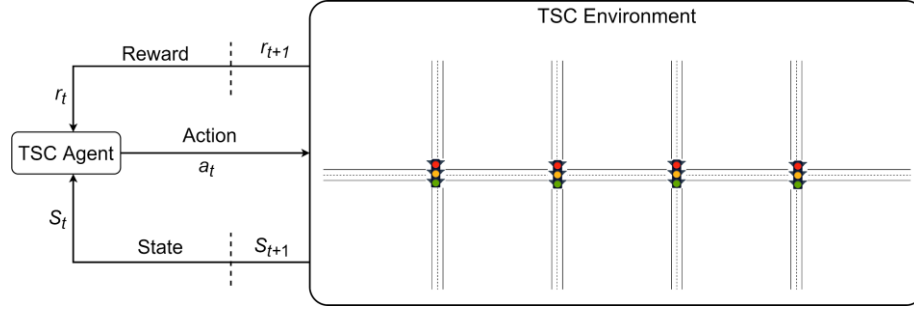


Figure 1. Single-agent RL architecture for regional traffic signal control.

The above framework highlights that RL-based TSC requires defining the following core components: state definition, action design, reward function formulation, environment model and strategy learning algorithm.

Table 1. Table of notations

Notations	Explanation
a	The action of the agent
S	The state observed by the agent
r	The reward obtained by the agent
q	The queue length of a link
t	The total travel time of all vehicles in the region
q_{ub}	The upper bound of queue length
L	The number of links in the region
s_{lb} and s_{ub}	The lower and upper bounds of the signal phase split
Δs	The phase split adjustment magnitude
M	The number of signal intersections in the region
q_{lc} and q_{hc}	The thresholds for light and heavy congestion, respectively
w_l	The link importance weight
w_{cp}	The penalty weight associated with heavy congestion
w_t	The travel time penalty weight

3.1 State

The state consists of three parts: the congestion state, the signal phase scheme and regional state representation.

Congestion state of a link:

The congestion state is quantified by the queue length q for each link (segment) at the current time, intrinsically associated with the physical road segment and enabling direct mapping to the monitored infrastructure. The queue length is constrained between 0 and q_{ub} , established beforehand.

Figure 2 below illustrates the spatio-temporal trajectories of a vehicle platoon that enters a specific road link and proceeds straight through both upstream and downstream

intersections during the same green light cycle (hereinafter referred to as the "through platoon"). It also depicts the queued vehicles at the downstream intersection. The queue length in this paper is defined as the number of queued vehicles. This queue length correlates strongly with the average travel time of the through platoon: Specifically, as queued vehicles consume green light time, longer queue lengths cause more vehicles in the through platoon to encounter red lights, thus increasing the average travel time. As such, queue length is an effective congestion metric for the road link.

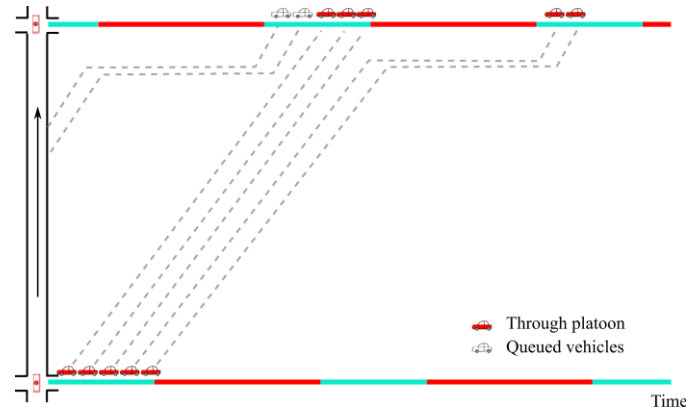


Figure 2. The definition of queue length

The rationale for adopting the queue length metric lies primarily in its estimability using minimal probe vehicle data. Previous studies proposed methods to estimate average link travel time (24) and queue length (12) with probe vehicle (floating car) data. Owing to space limitations, this paper omits these methods. In these methods, the probe vehicles (floating cars) constitute a sample of the through platoon. The input data consist solely of link travel times from these probe vehicles, using only aggregated travel times (without raw trajectory points).

Signal phase scheme:

The current signal phase scheme is defined as the signal phase split for each regional intersection, specifying the time allocation for traffic movements at this specific junction. The signal phase split is constrained between s_{lb} and s_{ub} , established beforehand.

In this study, each intersection is controlled by a four-phase signal (see Figure 3). It is assumed that the signal cycle, left-turn phase duration, yellow time, all-red time, and offset time are predetermined and fixed. The signal phase split is the only adjustable variable and defined as the sum of the north-south phases' time, which includes the north-south straight/right turn phase (Phase 1) and the left-turn phase (Phase 2). Given the signal cycle is constant, adjusting the signal phase split regulates traffic flow at upstream and downstream intersections, managing congestion levels. As shown in Figure 3, if congestion occurs on the link indicated by the arrow (from intersection m to $m + 1$), the congestion can be alleviated by reducing the number of vehicles entering this link by increasing the signal phase split at the upstream intersection (intersection m), or by increasing the number of vehicles exiting this link by decreasing the signal phase split at the downstream intersection (intersection $m + 1$).

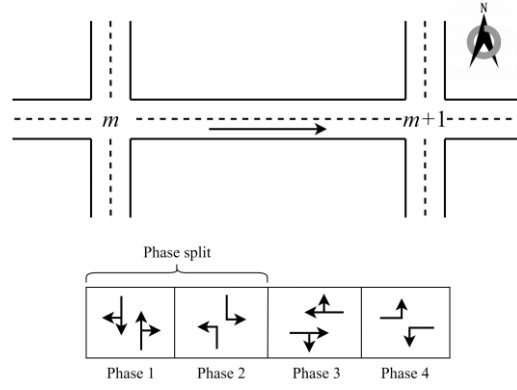


Figure 3. The signal phase split

Regional state representation:

The observed system state adopts an $M \times M$ adjacency matrix representation. Diagonal elements hold the signal timing parameters at each intersection, while off-diagonal elements capture the real-time traffic state metric q for physically connected road segments; all other entries are set to zero. For instance, as shown in Figure 4, in a 3×3 grid of intersections (9 nodes), the matrix dimension extends to 9×9 , where diagonal positions store signal parameters and off-diagonal elements contain queue length q for the physical links.

As a direct analogue to graph adjacency matrices, this representation explicitly maintains the topological connectivity of the transportation network via spatial element distribution. Non-zero entries storing q (e.g., row 1 column 2) correspond to physical links between intersections, while zero entries (e.g., row 1 column 7) directly indicate absent connections. The spatial distribution of q and zeros inherently preserves the network's connectivity graph – mapping intersections to nodes and non-zero q positions to edges.

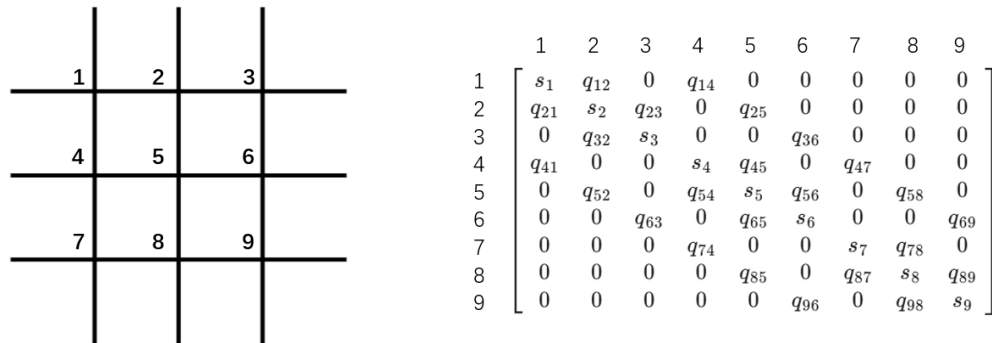


Figure 4. 3×3 Intersection State Visualization Diagram and 3×3 Intersection Grid Layout Diagram

The adjacency matrix is adopted for its intrinsic capability to unify signal parameters and traffic states q within a single framework. This framework inherently embeds connectivity relationships through the spatial distribution of non-zero and zero elements. This is illustrated by the 3×3 grid representation, where the arrangement of diagonal signal parameters and off-diagonal q values and zeros fully characterizes both intersection states and network topology.

3.2 Action

The action is structured in two stages: first, selecting an intersection, and then adjusting the phase split of that intersection.

For a chosen intersection, the agent has three possible actions: modifying the current signal phase split by $-\Delta s$, 0, and Δs , where Δs is a predefined value.

It should be noted that this adjustment is cumulative, and the phase split can ultimately be adjusted to the pre-set lower and upper bounds (s_{lb} and s_{ub}).

For M intersections, the size of the action space is $M \times 3$. This design ensures that as the number of intersections increases, the action space grows linearly and avoids explosive growth. In contrast, simultaneously adjusting the timing of all intersections would yield an action space of 3^M , which would cause an exponential explosion in size.

3.3 Reward

Queue length serves as the indicator of congestion level, and the reward is defined as a function of the queue length for each link in a region. To achieve the study's goal of congestion alleviation and total travel time minimization, a penalty weight is imposed during severe congestion.

The regional reward equals the sum of its component links' rewards, where each link's reward is defined by Eq. 1.

$$\begin{cases} q \leq q_{lc} \text{ (Free flow): } reward = 0 \\ q_{lc} \leq q \leq q_{hc} \text{ (Light congestion): } reward = -(w_l \times q) - (w_t \times t) \\ q \geq q_{hc} \text{ (Heavy congestion): } reward = -(w_{cp} \times w_l \times q) - (w_{cp} \times w_t \times t) \end{cases} \quad (1)$$

Where q is the queue length of a link, t represents the total travel time of all vehicles in the region, q_{lc} and q_{hc} are the thresholds for light and heavy congestion, respectively. w_l represents the link importance weight and w_{cp} is the penalty weight associated with heavy congestion, w_t denotes the travel time penalty weight. The values of q_{lc} , q_{hc} , w_l and w_{cp} are predetermined constants.

3.4 Environment

We implement the TSC environment using Gymnasium (25), an open-source Python library for developing and comparing reinforcement learning algorithms. Gymnasium facilitates seamless interaction between learning algorithms and environments through its standardized API.

The observation space adopts the adjacency matrix-like structure introduced previously. Each element within this matrix is normalized to the interval $[0, 1]$ based on its corresponding lower and upper bounds.

For the action space, all possible actions are discretized into integers, resulting in a total of $3 \times M$ integers.

The core components of the TSC environment consist of a traffic simulation model, a simulation controller and a traffic monitor. The simulation model must accurately replicate the

real-world region targeted for signal timing optimization, encompassing road geometries, intersection geometries, and traffic demand patterns. The simulation controller manages simulator operations, including running the simulation model and dispatching signal control schemes to the simulation model.

The traffic monitor collects the queue lengths of all road segments in the region. In this paper, queue lengths are estimated via data generated by all vehicles within the simulation platform, rather than through probe vehicle technology. This approach is feasible during the RL strategy's learning stage. However, during evaluation and deployment, probe vehicle technology should be adopted. To simplify the evaluation process, this paper employs the same approach employed in the training stage to estimate queue lengths. As queue length estimation based on probe vehicle data is an established method, this simplification has minimal impact on the model's evaluation.

At a specific control time (t), the queued vehicles on a given road segment comprise two parts, and the queue length can be determined using these criteria (see Figure).

Vehicles passing through this segment shall meet the following criteria: proceeding straight through the downstream intersection, with an entry time before the start time of the most recent green phase at the upstream intersection (t_{gs}^u) and a departure time after the start time of the next green phase at the downstream intersection (t_{gs}^d);

Vehicles remaining on this segment shall meet the following criteria: intended to proceed straight through the downstream intersection, with entry time before the start time of the most recent green phase at the upstream intersection (t_{gs}^u).

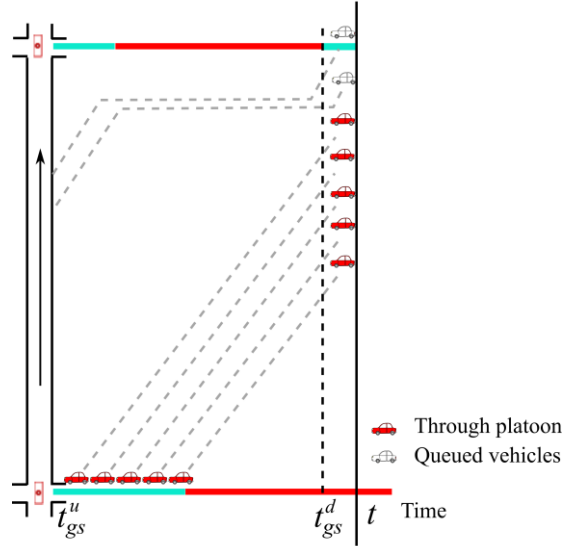


Figure 5. The queued vehicles on a road segment

3.5 DreamerV3

DreamerV3 was selected over PPO for policy training, cutting data and compute demands by dramatically reducing environment interaction versus traditional RL.

Recent world models excel at data-efficient learning in simulations and video games. Studies by Hafner et al. (27-29) validate this capability. DreamerV3 excels at data-efficient policy learning, as emphasized in (27).

DreamerV3 imagines action outcomes to cut real-world interaction, saving compute and time, while needing only two knobs—training ratio and model size—to boost data efficiency.

The DreamerV3 algorithm consists of two components: World Model Learning and Actor Critic Learning (27,29,30).

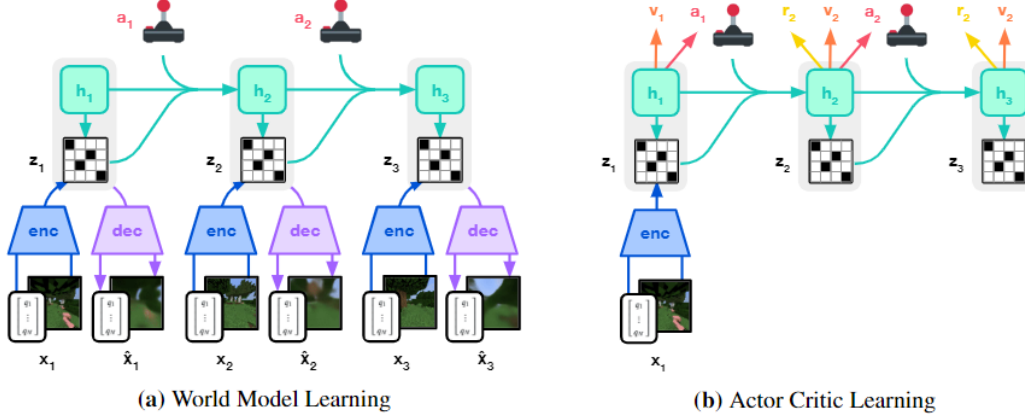


Figure 6. The DreamerV3 algorithm(27)

DreamerV3's Recurrent State-Space Model (RSSM) learns rich environment representations: encoder $x_t \rightarrow z_t$, recurrent state h_t predicts z_t from a_{t-1} ; combined (h_t, z_t) yields reward r_t , continuation $c_t \in \{0,1\}$, and reconstruction.

The actor and critic networks derive behaviors from imagined trajectories generated by the world model. The actor and critic operate on model states $s_t \doteq \{h_t, z_t\}$, leveraging the rich representations learned by the world model.

4. EXPERIMENT DESIGN

4.1 Simulation setup

The experiments utilize SUMO, a sophisticated and adaptable traffic simulation software, within a simulation environment. SUMO provides a dedicated interface, libsumo, enabling manipulation of the simulation via python scripts. Using the libsumo interface, the simulation controller can control simulation execution, access real-time traffic data, and dispatch signal schemes to SUMO. The mesoscopic mode is employed, computing vehicle movements that account for queues and operating 100 times faster than the microscopic mode.

The RL-based TSC method is tested in a region shown in Figure . Each link has three lanes and extends to four lanes at the downstream intersection. Each approach contains two straight lanes, one left-turn lane, and one right-turn lane. Each intersection is controlled by four signal phases, with their sequence predetermined. The signal cycle, left-turn phase time, yellow time, and all-red time are fixed at 100 seconds, 8 seconds, 2 seconds, and 2 seconds, respectively. The initial signal phase split is set to 50 seconds.

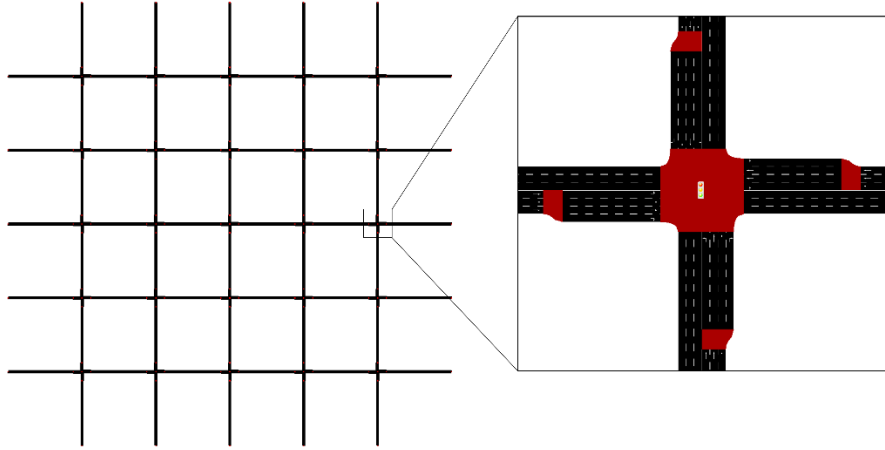


Figure 7. Region Diagram

Traffic is generated using zone-to-zone demand generation, in which Traffic Assignment Zones (TAZs) are utilized. Zone-to-Zone flows are defined by an origin-destination (OD) matrix.

The geometry of the region under study is depicted in Figure . Each link within this region has three lanes, widening to four lanes as it approaches the downstream intersection. This design mirrors the typical traffic infrastructure in many urban areas, to accommodate varying traffic volumes at intersections. Each approach at the intersection has two straight lanes, one left-turn lane, and one right-turn lane.

Every intersection in the region is managed by a four-phase signal system, and the order of these stages is predetermined. The time duration for the signal cycle, the left-turn stage time, the yellow light time, and the all-red light time are all fixed throughout the simulation. Specifically, they are set to 100 seconds, 8 seconds, 2 seconds, and 2 seconds respectively. The initial signal phase split is set to 50 seconds, which serves as a starting point for evaluating the effectiveness of the proposed traffic signal control model.

4.2 RL-model parameter setting

Table 2 lists the parameters of the RL-based TSC model. When the signal phase split is set to its initial value (50 seconds), the saturated flow per cycle for the straight lanes is 50 vehicles. The upper bound of the queue length is set to this level. This situation is considered extreme congestion because queued vehicles consume the entire green time, forcing all incoming vehicles to stop at the downstream intersection. The thresholds for light and heavy congestion (q_{lc} and q_{hc}) are set to 20% and 50% of this saturated flow level. In scenario 1, where the major traffic flows west to east, link-importance weights (w_l) are set to 0 for links in the opposing direction. This confines the reward calculation to links aligned with the main flow.

Table 2. RL-based TSC method parameters

Parameter	Value
q_{ub}	50 no. of vehicles
q_{lc} and q_{hc}	10 and 25 no. of vehicles (20 % and 50 % of q_{ub} , respectively)
s_{lb} and s_{ub}	30 s and 70 s
Δs	2 s
w_l	0 for the east-west bound links in the first pattern of the OD matrix; 1 for otherwise
w_{cp}	10

The RL policy network comprises two fully connected layers, each with 32 neurons.

The RL models are trained across multiple episodes, with each episode lasting 16,200 seconds, including a 1,800-second warm-up phase. During the warm-up, the signal scheme remains unchanged, and the vehicles are progressively introduced into the road network to ensure a stable initial state. Each episode contains 144 steps, determined by dividing the simulation time after the warm-up phase (16,200 - 1,800 seconds) by the control interval (t_c).

The RLLib library facilitates the learning of the signal control policy.

This research employs a high-performance computing platform equipped with an Intel Core i9-14900K processor, featuring 8 high-performance cores and 16 energy-efficient cores, enabling 32 threads in total. Additionally, the platform is integrated with an NVIDIA RTX A6000 48GB graphics card.

5.SIMULATION RESULTS AND DISCUSSIONS

5.1 Hyperparameter tuning for DreamerV3

For DreamerV3, two hyperparameters, specifically the training ratio and the model size, require adjustment. The training ratio denotes the ratio of replayed steps to environment steps. Hence, a higher training ratio typically enhances data efficiency significantly. As indicated in the literature (12), a model size of S is sufficient. Furthermore, a medium-scale training ratio substantially reduces hyperparameter-tuning time.

Figure shows the training curves of the model with a model size of S and training ratios of 64, 128, and 256, when the reward is set solely for congestion relief. To better distinguish final hyperparameter performance, the training curves after 36 hours are magnified. As shown in the figure, all three training ratios can complete the training within 48 hours. Also, as indicated in the DreamerV3 literature (13), the larger the training ratio, the faster the model achieves a higher episode reward.

Figure displays the training curves of the model with a model size of S and training ratios of 64, 128, and 256, when the reward aims to relieve congestion and minimize the total travel time. A similar trend can be observed. First, under the more complex reward setting, all three training ratios can complete the training within 48 hours. Second, generally omitted, the larger the training ratio, the faster the model tends to achieve a higher episode reward. However,

a deviation exists. When the training ratio is 64, the training curve exhibits pronounced fluctuations. Even towards the end, it continues to display intermittent large fluctuations.

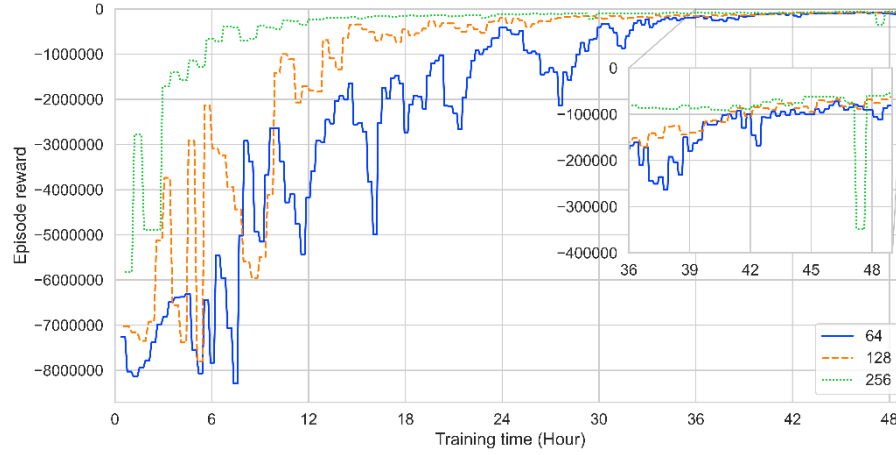


Figure 8 (a) Learning process (Scenario 1)

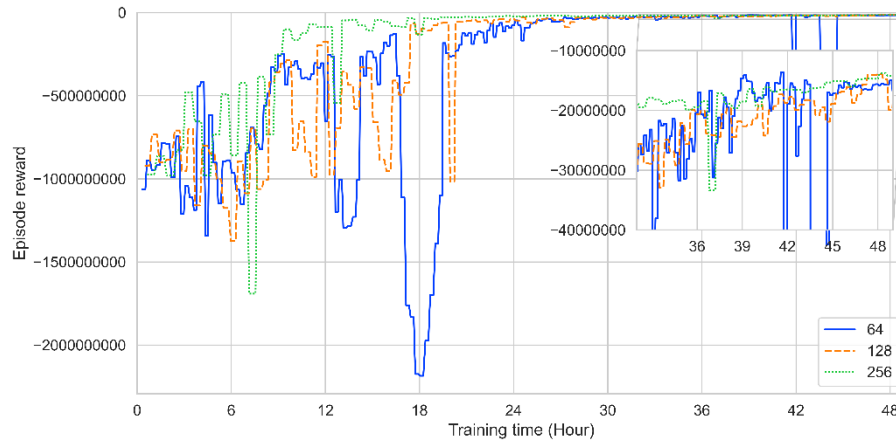


Figure 8(b). Learning process (Scenario 2)

5.2 Performance of TSC

The performance of the Traffic Signal Control (TSC) model was assessed through comparison against the baseline case, with a fixed signal-timing scheme. Evaluation occurred in two scenarios. In both scenarios, the TSC model was trained using the DreamerV3 algorithm. The model had a size of S and a training ratio of 128. In Strategy I, the reward was defined as congestion alleviation, whereas in Strategy II, it combined congestion alleviation with total travel time minimization.

Figure presents the distribution of queue lengths across road segments within one episode under Strategy I. The total queue length data points equal the product of the number of road segments and the number of signal cycles, yielding $80 \times 144 = 11,520$. As shown in the figure, under the baseline case, the queue length surpassed 50 on multiple occasions. Given that the through-traffic capacity of vehicles within one cycle is approximately 50, in these cases, all vehicles have to queue up due to the red light, indicating severe congestion. In contrast, under Scenario 1, the queue length remained below 40, and cases where it exceeded 30 were also

relatively few, indicating successful congestion control. Figure shows the queue length distribution under Strategy II, the results are similar to those in Figure and are therefore omitted for brevity.

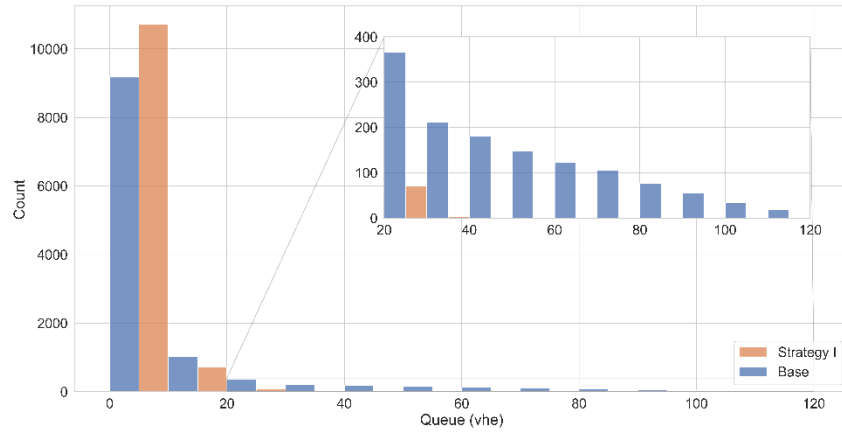


Figure 9(a). Queue length distribution when reward aims solely at congestion alleviation

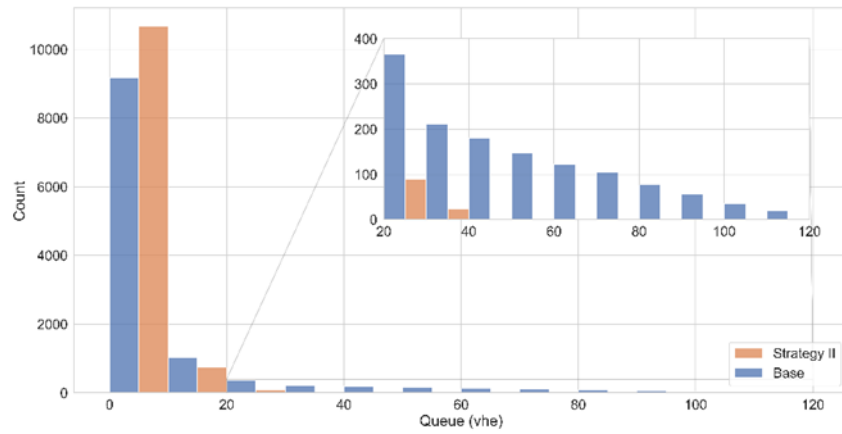


Figure 9(b). Queue length distribution when the reward seeks to alleviate congestion and minimize total travel time

Figure 10 compares the average travel time per episode under the baseline and under Strategy II. This average is computed as the total travel time divided by the number of vehicles. As shown in the figure, Strategy II significantly reduced the total travel time (to 63% of that in the base case). Currently, we cannot claim this is the minimum achievable travel time. We can confirm, however, that training converges successfully when the reward incorporates both congestion control and total travel time minimization.

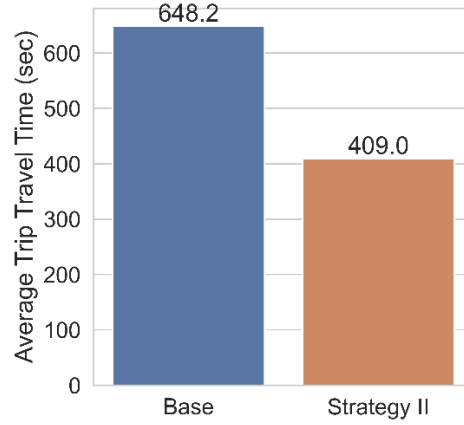


Figure 4. Total travel time comparison

6.CONCLUSION

This study introduces a single-agent RL model for TSC in urban regions, aiming to integrate probe vehicle technology for real-time congestion management. Unlike traditional multi-agent approaches, the model leverages a centralized paradigm, coordinating signal timings across multiple intersections.

The TSC model's efficiency arises from its queue length-driven state-reward-action design, establishing a direct link between congestion metrics and control actions. The state vector captures spatiotemporal queue profiles, while the reward function incorporates rewards for congestion alleviation and travel time minimization, prioritizing proactive queue dissipation. Actions adjust signal phase splits to regulate inflow/outflow, analogous to a feedback control system that significantly reduces queue variability relative to the baseline.

Simulation results obtained using the SUMO simulator across two traffic scenarios show that the TSC model significantly reduces queue lengths and average travel time, reaching 63% of the base case value by leveraging the DreamerV3 algorithm.

Future research will expand experiments to large-scale road networks with hundreds of intersections to validate scalability. Additionally, we will integrate graph neural networks and exploit the s-q transportation network interaction matrix to enhance data efficiency and policy learning in complex traffic environments.

DATA AVAILABILITY

The data that support the findings of this study are available from the author, Qiang Li (liqiangjl@qq.com), upon reasonable request.

CONFLICTS OF Interest

The authors declare that there are no conflicts of interest related to the publication of this paper.

ACKNOWLEDGMENTS

This study was supported by a grant from Guangdong Province General Regular Projects of Social Sciences Planning (No. GD24CGL37), Natural Science Foundation of Top Talent of SZTU (No. GDRC202322).

REFERENCES

1. H. Wei, G. Zheng, V. Gayah, and Z. Li, “A Survey on Traffic Signal Control Methods,” 2020, [Online]. Available: <http://arxiv.org/abs/1904.08117>
2. C. Wu, I. Kim, and Z. Ma, “Deep Reinforcement Learning Based Traffic Signal Control: A Comparative Analysis,” *Procedia Computer Science*, vol. 220, pp. 275–282, 2023, <https://doi.org/10.1016/j.procs.2023.03.036>.
3. K. Arulkumaran, M. P. Deisenroth, M. Brundage, and A. A. Bharath, “Deep Reinforcement Learning: A Brief Survey,” *IEEE Signal Processing Magazine*, vol. 34, no. 6, pp. 26–38, 2017, <https://doi.org/10.1109/MSP.2017.2743240>.
4. D. Li, J. Wu, M. Xu, Z. Wang, and K. Hu, “Adaptive Traffic Signal Control Model on Intersections Based on Deep Reinforcement Learning,” *Journal of Advanced Transportation*, vol. 2020, Aug. 2020, <https://doi.org/10.1155/2020/6505893>.
5. L. Zheng and B. Wu, “A Reinforcement Learning Based Traffic Control Strategy in a Macroscopic Fundamental Diagram Region,” *Journal of Advanced Transportation*, vol. 2022, Apr. 2022, <https://doi.org/10.1155/2022/5681234>.
6. A. Haydari and Y. Yilmaz, “Deep Reinforcement Learning for Intelligent Transportation Systems: A Survey,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 1, pp. 11–32, 2022, <https://doi.org/10.1109/TITS.2020.3008612>.
7. M. Noaeen *et al.*, “Reinforcement learning in urban network traffic signal control: A systematic literature review,” *Expert Systems with Applications*, vol. 199, p. 116830, Aug. 2022, <https://doi.org/10.1016/j.eswa.2022.116830>.
8. H. Wei and G. Zheng, “Recent Advances in Reinforcement Learning for Traffic Signal Control,” *ACM SIGKDD Explorations Newsletter*, vol. 22, pp. 12–18, 2021, [Online]. Available: <https://api.semanticscholar.org/CorpusID:229540095>.
9. H. Zhao, C. Dong, J. Cao, and Q. Chen, “A survey on deep reinforcement learning approaches for traffic signal control,” *Engineering Applications of Artificial Intelligence*, vol. 133, p. 108100, Jul. 2024, <https://doi.org/10.1016/j.engappai.2024.108100>.
10. C. Chen *et al.*, “Toward A Thousand Lights: Decentralized Deep Reinforcement Learning for Large-Scale Traffic Signal Control,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, Apr. 2020, pp. 3414–3421. <https://doi.org/10.1609/aaai.v34i04.5744>.
11. F. Lian, B. Chen, K. Zhang, L. Miao, J. Wu, and S. Luan, “Adaptive traffic signal control algorithms based on probe vehicle data,” *Journal of Intelligent Transportation Systems*, vol. 25, no. 1, pp. 41–57, Jan. 2021,

- <https://doi.org/10.1080/15472450.2020.1750384>.
12. Q. Li, H. Hu, and L. Miao, "Queue Length Estimation Using Probe Vehicle Data for a Congested Arterial Road," in *CICTP 2015 - Efficient, Safe, and Green Multimodal Transportation - Proceedings of the 15th COTA International Conference of Transportation Professionals*, Beijing, China, 2015, pp. 2295–2306. [Online]. Available: <http://dx.doi.org/10.1061/9780784479292.213>.
 13. J. J. Henry, J. L. Farges, and J. Tuffal, "The Prodyn Real Time Traffic Algorithm," *IFAC Proceedings Volumes*, vol. 16, no. 4, pp. 305–310, Apr. 1983, [https://doi.org/10.1016/S1474-6670\(17\)62577-1](https://doi.org/10.1016/S1474-6670(17)62577-1).
 14. H. Ceylan and M. G. H. Bell, "Traffic signal timing optimisation based on genetic algorithm approach, including drivers' routing," *Transportation Research Part B: Methodological*, vol. 38, no. 4, pp. 329–342, 2004, doi: [https://doi.org/10.1016/S0191-2615\(03\)00015-8](https://doi.org/10.1016/S0191-2615(03)00015-8).
 15. K. Pandit, D. Ghosal, H. M. Zhang, and C.-N. Chuah, "Adaptive Traffic Signal Control With Vehicular Ad hoc Networks," *IEEE Transactions on Vehicular Technology*, vol. 62, no. 4, pp. 1459–1471, 2013, <https://doi.org/10.1109/TVT.2013.2241460>.
 16. J. Little, M. Kelson, and N. Gartner, "MAXBAND: A Program for Setting Signals on Arteries and Triangular Networks," *Transportation Research Record Journal of the Transportation Research Board*, vol. 795, pp. 40–46, Dec. 1981, <https://dspace.mit.edu/handle/1721.1/1979>.
 17. K. Zhang, Z. Cui, and W. Ma, "A survey on reinforcement learning-based control for signalized intersections with connected automated vehicles," *Transport Reviews*, vol. 44, no. 6, pp. 1187–1208, Nov. 2024, <https://doi.org/10.1080/01441647.2024.2377637>.
 18. T. Chu, J. Wang, L. Codeca, and Z. Li, "Multi-Agent Deep Reinforcement Learning for Large-Scale Traffic Signal Control," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 3, pp. 1086–1095, Mar. 2020, <https://doi.org/10.1109/TITS.2019.2901791>.
 19. H. Wei *et al.*, "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*, Anchorage AK USA: ACM, Jul. 2019, pp. 1290–1298. <https://doi.org/10.1145/3292500.3330949>.
 20. P. Varaiya, "Max pressure control of a network of signalized intersections," *Transportation Research Part C: Emerging Technologies*, vol. 36, pp. 177–195, Nov. 2013, <https://doi.org/10.1016/j.trc.2013.08.014>.
 21. R. Bokade, X. Jin, and C. Amato, "Multi-Agent Reinforcement Learning Based on Representational Communication for Large-Scale Traffic Signal Control," *IEEE Access*, vol. 11, pp. 47646–47658, 2023, <https://doi.org/10.1109/ACCESS.2023.3275883>.
 22. L. Yan *et al.*, "Graph cooperation deep reinforcement learning for ecological urban

- traffic signal control,” *Applied Intelligence*, vol. 53, no. 6, pp. 6248–6265, Mar. 2023, <https://doi.org/10.1007/s10489-022-03208-w>.
23. J. Ma and F. Wu, “Learning to Coordinate Traffic Signals With Adaptive Network Partition,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 25, no. 1, pp. 263–274, Jan. 2024, <https://doi.org/10.1109/TITS.2023.3308594>.
 24. Q. Li, T. Miwa, and T. Morikawa, “Efficient link travel time estimation for signalized link by small size probe reports,” in *Proceedings of the 9th International Conference of Chinese Transportation Professionals, ICCTP 2009: Critical Issues in Transportation System Planning, Development, and Management*, Harbin, China, 2009, pp. 1884–1891. [Online]. Available: [http://dx.doi.org/10.1061/41064\(358\)265](http://dx.doi.org/10.1061/41064(358)265).
 25. M. Towers *et al.*, “Gymnasium: A Standard Interface for Reinforcement Learning Environments,” *arXiv preprint arXiv:2407.17032*, 2024. [Online]. Available: <https://arxiv.org/abs/2407.17032>.
 26. E. Liang *et al.*, “RLlib: Abstractions for Distributed Reinforcement Learning,” *arXiv preprint arXiv:1712.09381*, 2018. [Online]. Available: <https://arxiv.org/abs/1712.09381>.
 27. Hafner, D., J. Pasukonis, J. Ba, and T. Lillicrap, “Mastering Diverse Domains through World Models,” *arXiv preprint arXiv:2301.04104*, 2023. <https://doi.org/10.48550/arXiv.2301.04104>.
 28. Hafner, D., T. Lillicrap, M. Norouzi, and J. Ba, “Mastering Atari with Discrete World Models,” *arXiv preprint arXiv:2010.02193*, 2020. <https://doi.org/10.48550/arXiv.2010.02193>.
 29. Hafner, D., T. Lillicrap, J. Ba, and M. Norouzi, “Dream to Control: Learning Behaviors by Latent Imagination,” *arXiv preprint arXiv:1912.01603*, 2019. <https://doi.org/10.48550/arXiv.1912.01603>.
 30. Ha, D., and J. Schmidhuber, “World Models,” *arXiv preprint arXiv:1803.10122*, 2018. <https://doi.org/10.48550/arXiv.1803.10122>.