Spatial-Temporal Graph Convolutional Network for Large Scale Traffic Signal Control

Hanling Yi
hanling.cuhk@gmail.com
Intellifusion Inc.
Shenzhen, China

Xiaotian Yu
Intellifusion Inc.
Shenzhen, China
yu.xiaotian@intellif.com

Yuanfa Li
Intellifusion Inc.
Shenzhen, China
li.yuanfa@intellif.com

Xiancai Tian
Intellifusion Inc.
Shenzhen, China
mrtianxc@gmail.com

Yu Su
Intellifusion Inc.
Shenzhen, China
s641613918@gmail.com

ABSTRACT

Traffic congestion is becoming an increasingly critical issue to urban traffic management, it results in excess delays, reduced safety and is attributed to significant economic cost and inefficiency to the society as a whole. Strategical traffic signal control has been proved an efficient approach to mitigate such issue. The City Brain Challenge at KDD Cup 2021 invites participants to develop traffic signal control algorithms to maximize the number of vehicles served in a city-scale road network with real-world traffic demand, while maintaining an acceptable travel delay. The competition is based on a microscopic traffic simulation engine called CBEngine that provides traffic simulation on city-scale road networks. In this competition, we adopt the Proximal Policy Optimization algorithm with Clipped Objective (a.k.a. PPO), where a novel neural network structure is also proposed. In specific, we design a spatial-temporal graph convolutional network constituting both Gated Recurrent Units (GRU) and Graph Attention Networks (GAT) to capture comprehensive spatial-temporal correlation on traffic flows across the road network. The proposed model achieves competitive performance on the official flow dataset, in specific, 49,048 vehicles were served with a delay index of 1.4097.

CCS CONCEPTS

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

KEYWORDS

KDD Cup Challenge, Traffic Signal Control, Deep Reinforcement Learning, Proximal Policy Optimization

ACM Reference Format:
Hanling Yi, Xiaotian Yu, Yuanfa Li, Xiancai Tian, and Yu Su. 2021. Spatial-Temporal Graph Convolutional Network for Large Scale Traffic Signal Control. In *KDD Cup '21: ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, August 14–18, 2021, Virtual Conference.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/1122445.1122456

1 INTRODUCTION

As an increasing population flock to urban area, the problem of traffic congestion in these areas is becoming more serious than ever, which has lead to not only significant economic cost but also inefficiency to the society as a whole. Solutions that mitigate traffic congestion problem are highly demanded. As signalized intersections are one of the most prevalent bottleneck types in urban road networks, traffic signal control plays a vital role in urban traffic management. To improve the efficiency of traffic signal control, a variety of strategies have been proposed by researchers from both transportation community and data science community. In general, these strategies can mainly be classified into two categories, namely transportation approaches and deep learning-based methods.

Transportation approaches mainly adopt two kinds of strategies, i.e., fixed-time strategies and traffic-responsive strategies [1]. The fixed-time strategies for a given time of day (e.g., morning peak hour) are derived off-line based on historical constant demands and turning rates for each approach. Well-known examples of fixed-time strategies are SIGSET [2] and SIGCAP [3]. The main drawback of fixed-time strategies is that their settings are based on historical rather than real-time data. Since traffic condition can vary dramatically even in short duration, fixed-time strategies may fail to adapt for real time traffic patterns, and requires manual interference oftentimes. Meanwhile, traffic-responsive strategies make use of real-time traffic measures to choose the suitable signal settings in real time. For example, Max-pressure (MP) control [4] aims to reduce the risk of over-saturation by balancing queue length between neighboring intersections by minimizing the "pressure" of the phases for an intersection. Formally, the pressure of a movement signal is defined as the number of vehicles on incoming lanes (of the traffic movement) minus the number of vehicles on the corresponding outgoing lanes; the pressure of a phase is defined as the difference between the total queue length on incoming approaches and outgoing approaches. By setting the objective as minimizing the pressure of phases for individual intersections, MP is proved to maximize the throughput of the whole road network. [5] proposes an adaptive traffic light control algorithm that adjusts both the sequence and length of traffic lights in accordance with the real time traffic detected, considering a number of factors such as

KDD Cup '21, August 14–18, 2021, Virtual Conference

ACM ISBN 978-1-4503-XXXX-X/18/06…$15.00
https://doi.org/10.1145/1122445.1122456

traffic volume, waiting time, vehicle density. [6] proposes a stable longest queue first (LQF) signal scheduling algorithm that utilizes a maximal weight matching algorithm to minimize the queue sizes at each approach, yielding significantly lower average vehicle delay through the intersection. However, such methods have their limitations. One of the limitations is that they usually make strong assumptions about traffic flows. For example, [7] assumes that vehicles come in a uniform and constant rate. Such assumptions are usually oversimplified and deviate from real life traffic conditions.

In recent years, with the fast advancement of computing power and computational technologies, as well as the fast increasing availability of data, such as GPS data and video frames from street-facing surveillance cameras, new approaches for traffic signal control become possible. Among them, deep reinforcement learning (DRL) for traffic signal control has become a hot research topic in the intelligent transportation community [8–12]. The city-scale traffic signal control can be formulated as a multi-agent reinforcement learning problem, where each intersection with traffic light is modeled as an agent and multiple agents cooperate to maximize the throughput of the road network. Unlike transportation approaches, DRL methods directly learn from the observed data without making unrealistic assumptions about the model. It will learn and adjust the strategy based on the feedback from the environment, the model is dynamically learned through trial-and-error in the real environment. At present, the two most popular classes of deep reinforcement learning algorithms are value-based and policy-based methods, such as Deep Q-learning Network and Proximal Policy Optimization, respectively.

In this competition, we adopt the Proximal Policy Optimization algorithm with Clipped Objective (a.k.a. PPO) [13], which alternates between sampling data through interaction with the environment, and optimizing a "surrogate" objective function using stochastic gradient ascent. Whereas standard policy gradient methods perform one gradient update per data sample, PPO proposes a novel objective function that enables multiple epochs of minibatch updates. PPO has shown impressive performance on assorted benchmark tasks, including simulated robotic locomotion and Atari game playing, and overall strikes a favorable balance between sample complexity, simplicity, and wall-time. However, we note that in the original PPO method, the network structures in actor and critic networks are simple MLPs, which can not capture comprehensive spatial-temporal correlation in traffic data, especially in multi-agent setting where coordination between neighboring agents are essential for a good policy. To tackle this issue, we modify the network structure and propose a spatial-temporal graph convolutional network. Based on the experiment results, our proposed model achieves competitive performance on the official traffic flow dataset. In specific, when the delay index is constrained to be lower than 1.40, our model can serve 49,048 vehicles.

2 METHODS

In this section, we first formulate traffic light control problem as a reinforcement learning task by defining the state, action and reward function. We then present the novel network architectures that can capture comprehensive spatial-temporal correlation on traffic flows.

Table 1: Permissible actions on three-way intersections

Missing Approach	Permissible Actions
North	1, 4, 6
East	2, 3, 7
South	1, 4, 8
West	2, 3, 5

2.1 Agent Design

In the following, we introduce the state, action and reward design for an agent that controls the intersection.

States. The state is defined for one intersection. In CBEngine, a typical four-way intersection has 24 lanes, including 12 incoming lanes and 12 outgoing lanes as shown in Figure 1. For each vehicle v on the lane, we compute the remaining time t_r^v for the vehicle to reach the intersection based on its current position and speed. For each lane l, we focus on the vehicles with $t_r^v \le 10$ and denote this set of vehicles as V_l. We record the number of time steps t_s^v that vehicle v stays on the current lane. Intuitively, large t_s^v means that the vehicle v has stayed on the current lane for a long time, and it should has high priority to pass the intersection. We then define virtual queue length of a lane l as

$$q^l = \sum_{v \in V_l} \left(1 + \max(0, t_s^v - \bar{t}_s) * \bar{w}\right), \tag{1}$$

where $\bar{t}_s$ and $\bar{w}$ are two hyper parameters, and in our model we set $\bar{t}_s = 6, \bar{w} = 0.1$. Compared to the queue length that simply count the number of vehicles on each lane, the virtual queue length defined in (1) takes into consideration the number of time steps that the vehicle stay in the current lane. By definition, the virtual queue length will be larger if the lane has more longstanding vehicles.

The state of an intersection s is then defined as a vector of the virtual queue length of 24 lanes, concatenated with a one-hot encoding vector for the current phase. We have $s \in \mathbf{R}^{32}$.

Actions. For a traffic signal, there are at most 8 phases. Each phase allows a pair of non-conflict traffic movement to pass this intersection. At time t, each agent chooses a phase p as its action a_t from action set $A \in \{1, 2, 3, 4, 5, 6, 7, 8\}$, indicating the traffic signal should be set to phase p. In our setting, a four-way intersection has eight permissible actions, correspondingly eight phases in Figure 1. Meanwhile, a three-way intersection has three permissible actions. Depending on the missing approach, the permissible actions on three-way intersection may be different, as shown in Table 1. For example, in a three-way intersection without the north approach, only phases 1, 4, 6 are permissible.

Rewards. We define rewards as throughput of the intersection. Specifically, for intersection i, we denote V_t^i as the set of vehicles that pass through it at time t. Then the reward of intersection can be computed as

$$r_t^i = \sum_{v \in V_t^i} \left(1 + \max(0, t_s^v - \bar{t}_s) * \bar{w}\right). \tag{2}$$

Similar to the definition of virtual queue length in (1), the reward defined in (2) considers the number of time steps that vehicle stays

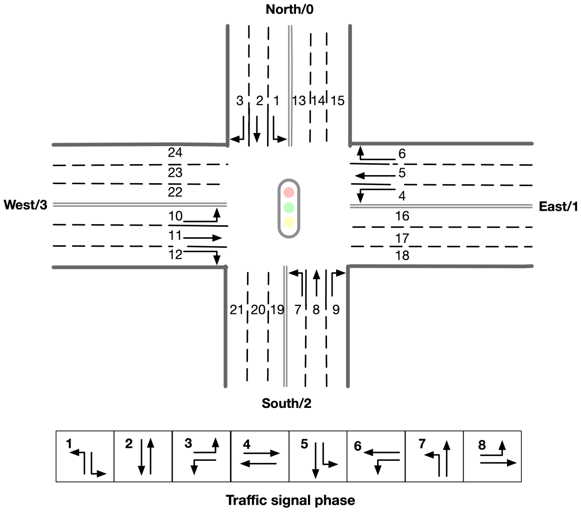

Figure 1: Illustration of the traffic movements and signal phase for a typical four-way intersection. If an agent is at phase 1, lane 1 and lane 7 along with all right turning lanes are passable.

on the lane, which intuitively encourages the agent to focus more on the lanes with longstanding vehicles.

2.2 Network Architecture

We adopt PPO to solve the traffic signal control problem. As an actor-critic based policy gradient approach, it consists of two networks: the actor network and the critic network. In the following, we will present the details of these two network architectures.

Actor Network. The actor network takes current state and recent historical states of the intersection as input, and outputs the probability distribution of actions for the intersection. As shown in Figure 2, the actor network consists of a GRU layer and multiple fully connected layers with ReLU activations. The GRU layer is introduced to learn the temporal dynamics in the traffic flow. With GRU layer, the actor network can better capture the temporal dependencies that naturally exists in the traffic flow.

Meanwhile, we note that there is a masking layer at the end of the actor network. This layer is specially designed to handle the three-way intersections. Specifically, we add a mask on the permissible actions for three-way intersections (shown in Table 1) before the softmax function, and output probability distribution on the three permissible actions for three-way intersections.

Following the standard PPO setting, in our implementation, we maintain two policy networks. The first one is the current policy that we want to refine, denoted as $\pi_\theta(a_t|s_t)$. The second is the policy that we used to collect samples, denoted as $\pi_{\theta_k}(a_t|s_t)$. With clipped objective, we compute a ratio between the new policy and the old policy as

$$r_t(\theta) = \pi_\theta(a_t|s_t)/\pi_{\theta_k}(a_t|s_t).$$

The clipped objective function is then defined as follows.

$$\mathcal{L}_{Clip}(\theta) = \mathbf{E}_t[\min(r_t(\theta)A_t, clip(r_t(\theta), 1-\epsilon, 1+\epsilon))A_t], \quad (3)$$

where A_t is an estimate of the advantage function at step t. Meanwhile, we add an entropy term in the loss function to encourage exploration during training. In summary, the loss function of the actor network can be expressed as follows:

$$\mathcal{L}_{Actor}(\theta) = \mathcal{L}_{Clip}(\theta) + \alpha\mathcal{L}_{Entropy}(\theta), \quad (4)$$

where $\mathcal{L}_{Entropy}(\theta)$ is the entropy term and α is a hyper parameter.

Critic Network. The critic network is a variant of spatial temporal graph convolutional network [14, 15]. It consists of a GAT layer, a GRU layer and multiple fully connected layers with ReLU activations, as shown in Figure 2. The GAT layer takes a graph $\mathcal{G}$ as input, which encodes spatial dependencies among agents. In $\mathcal{G}$, each node represents one agent. Each agent has edges connecting to itself and its nearest $K = 4$ neighboring agents [1]. By this definition, graph $\mathcal{G}$ has 1004 nodes and 5894 edges. Intuitively, the introduction of GAT layer in critic network allows message-passing between neighboring agents, thus encourages communication in the multi-agents setting. Meanwhile, the GRU layer allows the agent to better capture temporal dynamics in the traffic flow. By combining GAT and GRU, the resulting critic network can better capture the comprehensive spatial-temporal correlation on traffic flows.

3 EXPERIMENTS

In this section, we first introduce the experimental settings in the simulation. Then we present the details of training process. Finally, we present the experimental results.

3.1 Experimental Settings

Road Network and Traffic Flow. We use the road network in the final round and the default flow for simulation. There are in total 2,067 intersections with 3,041 roads in the road network, among which 1,004 of the intersections have traffic lights and 497 of them are three-way intersections. We run the simulation for 1,200 seconds. In the simulation, 74,926 vehicles enter the road network according to the time and route defined in flow file.

Evaluation Metric. We follow the competition rules and use the total number of vehicles served (i.e., total number of vehicles entering the road network) and delay index calculated every 20 seconds for model evaluation. The evaluation process is terminated once the delay index exceeds the predefined threshold 1.40.

The delay index is defined as the average delay index over all vehicles served in the network. For each vehicle, its delay index is computed as actual travel time divided by travel time at free-flow speed. For an uncompleted trip, the free-flow speed is used to estimate the travel time of rest of the trip. The delay index can be formally expressed as:

$$D = \frac{1}{N}\sum_{i=1}^{N}\frac{TT_i + TT_i^r}{TT_i^f}, \quad (5)$$

where N is the totol number of vehicles, TT_i is travel time of vehicle, TT_i^r is the rest of trip travel time estimated with free flow speed and TT_i^f is the full trip travel time at free-flow speed.

[1] We measure the distance between two agents using the length of shortest path between these two agents on the road network.

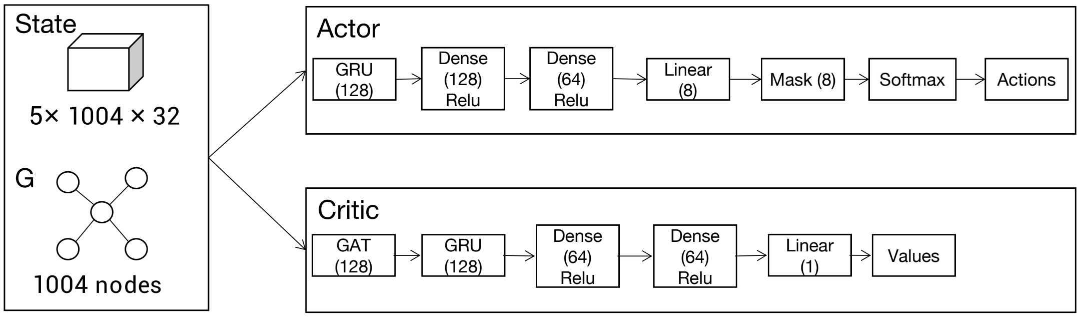

Figure 2: Illustration of the network architecture.

Table 2: Parameter settings

Parameter	Value	Description
a_{lr}	0.0001	Actor learning rate
c_{lr}	0.0001	Critic learning rate
γ	0.99	Discounted rate
ϵ	0.2	Trust region
α	0.01	Weight of entropy loss
k	1	Sample reuse times
h	5	Length of historical states

Discussion. In the simulation environment, we observe that some vehicles suddenly stopped on the lane with no reason. Depending on the position of the stopped vehicle, it may cause congestion on the intersection because the stopped vehicle will prevent other vehicles to pass through the intersection. This randomness in the driving behaviors of different vehicles bring additional challenge for our model.

3.2 Training Process

We implement the model from scratch using Pytorch. Both actor and critic networks are trained using Adam optimizer. During each update iteration, we train the actor and critic network alternatively. The two policy networks $\pi_\theta(a_t|s_t)$ and $\pi_{\theta_k}(a_t|s_t)$ are synchronized for every $T = 5$ updates. Multiple traffic flows are used to train the model. Specifically, we randomly generate 5 traffic flows. We use the official traffic flow to train the network for 500 episodes and the rest to finetune the network. The hyperparameters for training our model are summarized in table 2.

3.3 Experimental Results

We compared the served vehicle number and delay index of our model with other benchmarks as follows:

- **Fixed Time (FT)**. FT algorithm chooses a phase from a predefined cyclic phase sequence for every 20 seconds.
- **Max Pressure (MP)**. MP is a state-of-the-art network-level traffic signal control method, which greedily chooses the phase with the maximum pressure.
- **Longest Queue First (LQF)**. LQF is another greedy algorithm that chooses the phase with the longest virtual queue length.

Table 3: Performance of benchmarks

Model	Delay Index	Served Vehicle Number
FT	1.4033	37672
MP	1.4052	43688
LQF	1.4038	47747
DQN	1.4087	48590
PPO	1.4097	**49048**

- **Deep Q-learning Network (DQN)**. A DRL based traffic signal control method that uses the same network structure in the critic network as the base model.

Note that the first three algorithms are naive baselines that do not require any training. DQN is a value-based DRL method that uses a deep neural network to approximate the Q values. The results of all the baselines are shown in table 3. We can observe that PPO serves the most vehicles among all the baselines.

REFERENCES

[1] Markos Papageorgiou, Christina Diakaki, Vaya Dinopoulou, Apostolos Kotsialos, and Yibing Wang. Review of road traffic control strategies. *Proceedings of the IEEE*, 91(12):2043–2067, 2003.
[2] RB Allsop. Sigset: a computer program for calculating traffic capacity of signal-controlled road. *Traffic Eng Control*, 12:58–60, 1971.
[3] Fo Vo Webster. Traffic signal settings. Technical report, 1958.
[4] Pravin Varaiya. The max-pressure controller for arbitrary networks of signalized intersections. In *Advances in Dynamic Network Modeling in Complex Transportation Systems*, pages 27–66. Springer, 2013.
[5] Binbin Zhou, Jiannong Cao, Xiaoqin Zeng, and Hejun Wu. Adaptive traffic light control in wireless sensor network-based intelligent transportation system. In *2010 IEEE 72nd Vehicular technology conference-fall*, pages 1–5. IEEE, 2010.
[6] R Wunderlich, I Elhanany, and T Urbanik. A stable longest queue first signal scheduling algorithm for an isolated intersection. In *2007 IEEE International Conference on Vehicular Electronics and Safety*, pages 1–6. IEEE, 2007.
[7] Roger P Roess, Elena S Prassas, and William R McShane. *Traffic engineering*. Pearson/Prentice Hall, 2004.
[8] Hua Wei, Chacha Chen, Guanjie Zheng, Kan Wu, Vikash Gayah, Kai Xu, and Zhenhui Li. Presslight: Learning max pressure control to coordinate traffic signals in arterial network. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1290–1298, 2019.
[9] Hua Wei, Nan Xu, Huichu Zhang, Guanjie Zheng, Xinshi Zang, Chacha Chen, Weinan Zhang, Yanmin Zhu, Kai Xu, and Zhenhui Li. Colight: Learning network-level cooperation for traffic signal control. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pages 1913–1922, 2019.
[10] Elise Van der Pol and Frans A Oliehoek. Coordinated deep reinforcement learners for traffic light control. *Proceedings of Learning, Inference and Control of Multi-Agent Systems (at NIPS 2016)*, 2016.
[11] Mohammad Aslani, Stefan Seipel, Mohammad Saadi Mesgari, and Marco Wiering. Traffic signal optimization through discrete and continuous reinforcement

learning with robustness analysis in downtown tehran. *Advanced Engineering Informatics*, 38:639–655, 2018.

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

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

[14] Bing Yu, Haoteng Yin, and Zhanxing Zhu. Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. *arXiv preprint arXiv:1709.04875*, 2017.

[15] Shengnan Guo, Youfang Lin, Ning Feng, Chao Song, and Huaiyu Wan. Attention based spatial-temporal graph convolutional networks for traffic flow forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 922–929, 2019.