# Revisiting Pixel-based Traffic Signal Controls using Reinforcement Learning with World Models

**Toan V. Tran and Mina Sartipi**
Center for Urban Informatics and Progress
University of Tennessee at Chattanooga
Tennessee, USA

## Abstract

Improving Traffic Signal Control (TSC) can bring many benefits such as saving time for drivers and reducing emission discharged by vehicles. Many studies have proposed reinforcement learning-based traffic signal controls which outperform traditional approaches from traffic engineering. Generally, reinforcement learning for TSC can be categorized into feature-based, pixel-based, and hybrid methods. Regarding to performance, feature-based TSCs outperform the other kinds. In this paper, we revisit pixel-based TSCs. We propose *WorldLight* that learns a representation of traffic-state images and inputs the representation into a reinforcement learning controller. *WorldLight* not only closes the gap between pixel-based and feature-based methods but also outperforms state-of-the-art methods in some scenarios.

## Introduction

Traffic congestion is still a major problem to many cities around the world. It is because despite the massive investments in infrastructure to enhance the transportation system, the increase in mobility demand within the metropolitan area outstrips the network's capacity. Enormous congestion costs include the loss of productivity, pollution and environmental damages, and poor health due to stress and accidents while commuting. According to the Urban Mobility report (David Schrank 2021), traffic congestion causes 4.3 billion hours of delay, wastes 101 billion gallons of fuel, and damages the U.S. economy around $101 billion USD in 2021. One of the simplest yet effective methods to combat congestion is the use of traffic signal control (TSC) which aims to resolve conflict among different traffic movements and minimize vehicle travel time at the intersections. Optimizing TSC to improve traffic flows and reduce congestion has been intensively researched but is still a very active research area due to the emerging of new technology and better accessibility of traffic data.

Due to its importance, TSC has been studied for a long time. Generally, there are two approaches: rule-based and learned controls. More specifically, the early research (James Bonneson 2011; Varaiya 2013) proposed rule-based TSCs that are manually designed based on traffic-flow theories. An example of a rule-based control is the actuated controller (James Bonneson 2011), which has been widely deployed in the field. These controllers read real-time data from detectors to make actions. However, the traffic-flow theories are usually developed with unrealistic assumptions which can result in non-optimal solutions for the real-world environments. With the development of machine learning, some researchers have applied reinforcement learning (RL) for TSC (Wei et al. 2018; Liang et al. 2019; van der Pol and Oliehoek 2016). By learning via a long trial-and-error process, RL agents outperformed rule-based TSCs (Ault and Sharon 2021; Tran, Doan, and Sartipi 2021; Mei et al. 2022).

In the scope of this paper, we focus on investigating RL-based TSCs. In general, RL for TSC can be categorized into: pixel-based, feature-based, and hybrid methods. The first studies that applied RL for TSC are pixel-based methods (Liang et al. 2019; van der Pol and Oliehoek 2016), which directly apply traffic-state images to the input. To understand such high-dimensional data (i.e., images), pixel-based TSCs implement convolutional neural networks, illustrated in Figure 1. Other methods that are feature-based (Zheng et al. 2019; Chen et al. 2020; Wei et al. 2019) exact features such as number of vehicles on each lane, queue lengths, and average speed of vehicles; then the RL controller inputs these features instead of the image iteself. Figure 2 presents the workflow of feature-based RL for TSCs. There are also hybrid approaches that use both traffic-state images and features as inputs to the RL controller.

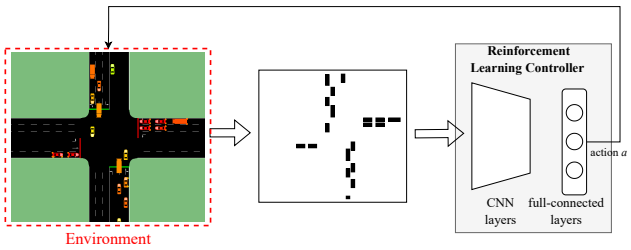



Figure 1: Pixel-based reinforcement learning for TSC

This paper is supported partially by the DOE DE-EE0009208 and NSF CCRI-2120358. The views and opinions of authors expressed herein do not necessarily state or reflect those of the U.S. Government or any agency thereof.

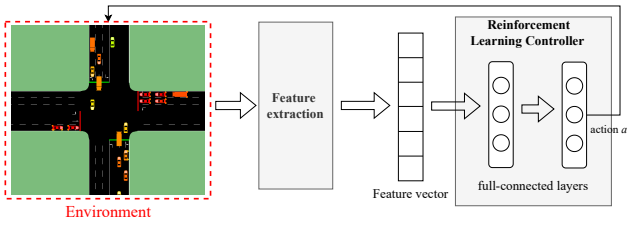



Figure 2: Feature-based reinforcement learning for TSC

The high-dimensional images are difficult for RL to learn during the trial-and-error process which has helped the pixel-based TSCs to outperform both feature-based and hybrid methods (Ault and Sharon 2021; Tran, Doan, and Sartipi 2021). Although the feature-based TSCs are state-of-the-art methods, they have some limitations. First, feature-based state representation methods are not optimal because they lack comprehensive information. Second, it requires manual work for feature designing. Finally, each existing feature design is for a particular reward function. For example, to optimize queue lengths (Zheng et al. 2019), the authors had to try different feature designs to identify that the number of vehicles on each lane has most influences.

In this paper, we revisit the pixel-based TSCs. We propose *WorldLight* that can not only achieve competitive performance compared to feature-based TSCs, but it also outperforms them in some scenarios. Instead of directly learning from traffic-state images during trial-and-error processes, *WorldLight* implements a world model (Ha and Schmidhuber 2018a; 2018b) to learn the representation of the images. The learned representation of *WorldLight* is comprehensive, since it includes information on both the current and future traffic states. In general, our contribution can be summarized as the follows. We propose *WorldLight* that outperforms state-of-the-art methods in some scenarios. The methodology of *WorldLight*, i.e., learning the representation of traffic-state images, closes the gap between pixel-based and feature-based TSCs. We also conduct some analysis for *WorldLight* including reward functions and RL algorithms.

The rest of this paper is organized as follows: Section  introduces *WorldLight* and details on the training process. Experiments and results are summarized in . Finally, Section  concludes the paper and gives an outlook on future work.

## Proposed Methodology – WorldLight

Figure 3 presents the architecture of the proposed method. *WordLight* is a pixel-based method. The input of *WorldLight* is an image which can represent comprehensive information of the current traffic status. To extract features from the image, *WordLight* implements a world model (Ha and Schmidhuber 2018a; 2018b). The world model includes two main components: AutoEncoder (AE) and Recurrent Mixture Density Network (RMDN). More specifically, AE is to represent the image into a latent vector $z$ whose dimension is much smaller than the raw image's. Meanwhile, RMDN aims to represent traffic modeling. RMDN inputs the latent vector $z_t$ at time $t$ and tries to predict the next latent vector $z_{t+1}$ at time $t+1$. More precisely, RMDN includes RNN layers to model the time series and a Mixture Density Network (MDN) to understand the uncertainty of traffic. Subsequently, the state vector $s$ created by the world model is the combination of the latent vector $z$ and the hidden state $h$. This state includes the information related to not only the current traffic status (i.e., $z$), but also the traffic modeling/prediction (i.e., $h$). Finally, the RL controller inputs the state vector $s$ and returns the optimal action $a$. Because the raw image is extracted to features by the world model, the RL controller implements two full connected layers.

- **AutoEncoder**: We implement a variational autoencoder. The autoencoder compresses high-dimensional traffic-state images to small latent vectors while minimizing information lost. The constructed image is more accurate while increasing the latent vector size. However, the larger latent vector causes more difficult for the RL controller.
- **Recurrent Mixture Density Network**: RMDN aims to do the modeling task $P(z_{t+1}|a_t, z_t, h_t)$. There are two components in RMDN: Recurrent Neural Network (RNN) and Mixture Density Network (MDN) (Bishop 1994). More specifically, RMDN applies LSTM layers (Hochreiter and Schmidhuber 1997) for the RNN part to understand time series patterns. Meanwhile, MDN is to model traffic uncertainty. Instead of outputting directly $z_{t+1}$, MDN returns $K$ Gaussian distributions for each element of $z_{t+1}$. More precisely, outputs of distribution $k$ ($\text{dist}_k$) include mixing coefficient $\pi_k$, standard deviation $\sigma_k$, and mean $\mu_k$. All of these outputs are from the last full connected layer of RMDN. Therefore, the number of units of the last layers is $|z| * K * 3$. RMDN's probability density function $p(x)$ is shown in Equation 1.

$$p(x) = \sum_{k}^{K} \pi_k \mathcal{N}(x|\mu_k, \sigma^2) \tag{1}$$

From the probability density function, we can train RMDN by minizing the log likelihood loss function $\mathcal{L}(w)$.

$$\mathcal{L}(w) = -\sum_{i=0}^{|z|} \ln \left( \sum_{k=1}^{K} \pi_k(w) \mathcal{N}\left(z_i|\mu_k(w), \sigma_k^2(w)\right) \right) \tag{2}$$

- **Reinforcement Learning Controller**: Because the input of the controller is a 1D vector, the controller implements a standard full-connected neural network including 2 hidden layers with 64 units. To train this neural network, we use the Proximal Policy Optimization (PPO) algorithm (Schulman et al. 2017) which is the most powerful policy gradient method for RL. RL agent includes three components: state, reward, and action. As mentioned above, the *WordLight*'s state is a vector which is the output of the world model. The reward is the total negative queue lengths of incoming lanes. The action is the phase index which will be executed for each interval $\tau$. Generally, for each interval, the world model processes the observed traffic-state image to state vector $s$. Subsequently, the controller uses this vector to predict the optimal phase which will be executed for the interval.

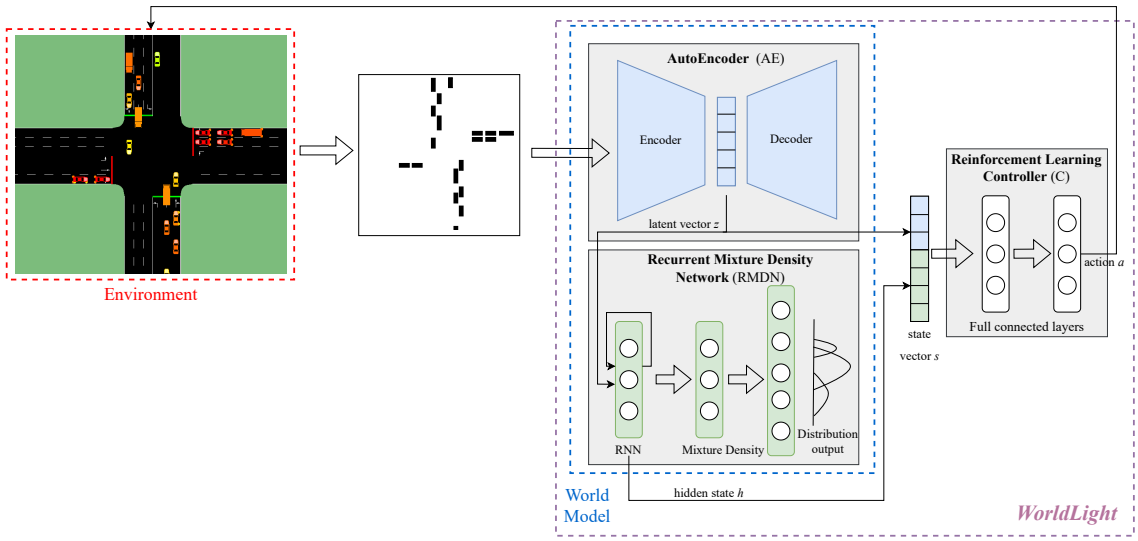



Figure 3: *WorldLight*'s architecture

### Training *WorldLight*

The training process for *WorldLight* is summarized as the follows:

1. Collect rollouts from a random policy/actuated control.
2. Train AE.
3. Train RMDN to model $P(z_{t+1}|a_t, z_t, h_t)$.
4. Train RL Controller via a trial-and-error process.

## Experiments & Results

### Simulation setting

We conduct experiments in SUMO, which is a popular simulation in transportation. All experiments in this paper simulate a real-world intersection – MLK Blvd & Market St, Chattanooga, TN, USA (Harris, Stovall, and Sartipi 2019). The traffic demand is from this corridor and similar to several existing datasets, provides number of vehicles. Additionally, it provides the arrival time of each vehicle, vehicle movement vehicle class, and vehicle length. By using this dataset, we can provide a realistic simulation.

### Baselines

• *LIT* (Zheng et al. 2019) is a feature-based deep Q learning method with a careful design of state and reward.

• *MPLight* (Chen et al. 2020) is also a feature-based method using deep Q learning.

• *CNN-L* (van der Pol and Oliehoek 2016) is a pixel-based RL method which directly inputs an image of the intersection as the state.

• *Actuated* (James Bonneson 2011) is a rule-based method that are running in the real world.

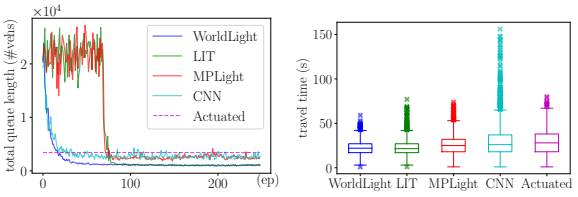
### EXP1: Overall performance of *WorldLight*

• **Setting**: We conduct this experiment for a single intersection during the the peak hour from 4pm to 5pm. As stated before, data is collected from the MLK Smart Corridor, so our simulation is representing the exact distribution of traffic flow. Each method is trained for 250 episodes.

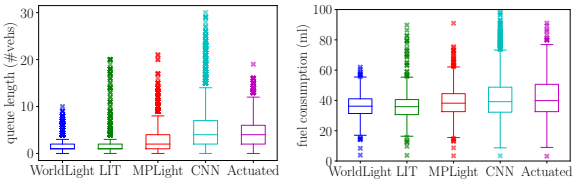
• **Result**: Figure 4 presents the overall performance of each methods. Generally, all RL-based methods outperform the *Actuated* control. Feature-based RL methods (i.e., *LIT* and *MPLight*) outperform the traditional pixel-based RL – *CNN-L*. Although *WorldLight* is a pixel-based method, by using world models, it outperforms the feature-based methods. Figure 4a presents the total queue length during the training process. *WorldLight* and *CNN-L* converge faster than *LIT* and *MPLight*, as illustrated in Figure 4a. One justification for this can be the different reinforcement learning algorithms (i.e., *WorldLight* and *CNN-L* use Proximal Policy Optimization (Schulman et al. 2017), *LIT* and *MPLight* utilize Deep Q Learning (Mnih et al. 2015)). In addition, Figures 4b, 4c, and 4d show the average travel time, queue length, and fuel consumption in the testing simulation. More precisely, *WorldLight* slightly outperforms *LIT* and provides more-stable performance across vehicles, i.e., smaller outlier values and less number of outliers. *Actuated* is also stable because it is a rule-based strategy, while *CNN-L* can cause long delay for some vehicles.

### EXP2: Effects of reward function

• **Setting**: In this experiment, we investigate performance of state representations of methods with different reward functions. The simulation setting remains the same as the one in Experiment 1 and each model is trained for 250 episodes. We considered four reward functions: one lane-based, one movement-based, and two vehicle-based. $r_1$ and $r_2$ are vehicle-based reward functions that consider waiting time and average speed, respectively. Subsequently, $r_3$ is a lane-based reward function considering queue lengths. Finally, the movement-based reward function $r_4$ focuses on pressure.




(a) Total queue length during training (b) Average travel time of vehicles during testing




(c) Average queue length of steps during testing (d) Average fuel consumption of vehicles during testing

Figure 4: Performance of *WorldLight* and previous methods

• **Result**: Figure 5 depicts reward values of methods. For all reward functions, the feature-based methods outperform the traditional pixel-based methods. This trend is similar to Experiment 1 as well as the results from previous studies (Wei et al. 2021). However, our method with world models has closed the gap between feature-based and pixel-based strategies. *WorldLight* performs the best in cases of $r_3$ (queue length) and $r_4$ (pressure). On the other hand, *LIT* is better than *WorldLight* for $r_1$ (waiting time) and $r_2$ (speed). This experiment shows that there is still no optimal way of state representation that works for all reward functions. This finding is also consistent with those in (Egea and Connaughton 2021). Therefore, designing state representation is a critical task which strongly depends to the objective (i.e., reward function).

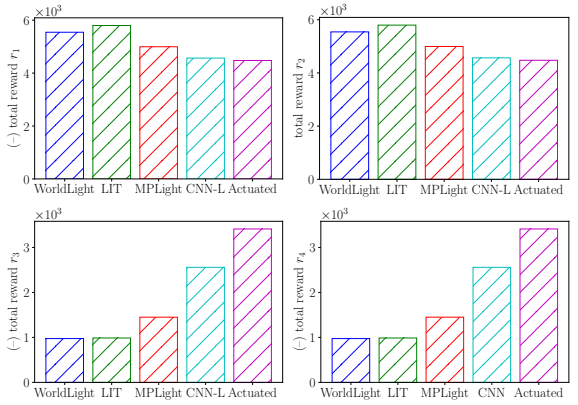
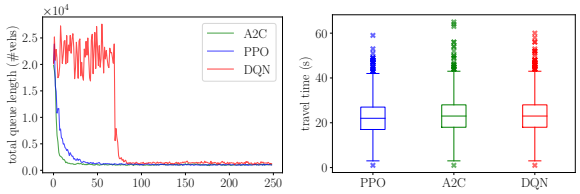



Figure 5: Performance of the state representation methods when using the same reward functions. For the negative total reward figures (i.e., $r1$, $r2$, and $r4$), the lower value is better. On the other hand, for the figure about $r3$, the higher value is better.

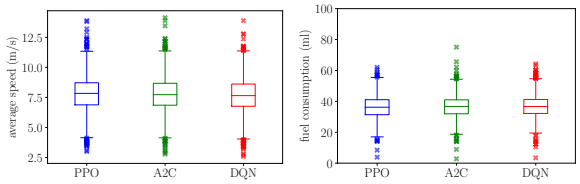
## EXP3: Effects of RL algorithms

• **Setting**: In this experiment, we investigate *WorldLight*'s performance when using various reinforcement learning algorithms. The simulation setting stays the same to the previous experiments. In total, we investigate three common RL algorithms including Proximal Policy Optimization (PPO) (Schulman et al. 2017), Advantage Actor Critic (A2C) (Mnih et al. 2016), and Deep Q Learning (DQN) (Mnih et al. 2015).

• **Result**: Figure 6 shows the performance of *WorldLight* using different RL algorithms. More specifically, A2C is the fastest training method, which converges after around 20 episodes. Meanwhile, PPO and DQN require 50 and 80 episodes, respectively. For the testing phase, PPO outperforms A2C and DQN. For instance, PPO reduces the average travel time of vehicles by 0.9% and 2.3% compared to A2C and DQN respectively. Futhermore, PPO rises the average speed by 0.6% and 1.9% compared to A2C and DQN. This experiment demonstrates the policy optimization methods (i.e., PPO and A2C) are better than the Q-learning method for *WorldLight*.




(a) Total queue length during training (b) Average travel time of vehicles during testing




(c) Average speed of vehicles during testing (d) Average fuel consumption of vehicles during testing

Figure 6: Performance of *WorldLight* when using different RL algorithms for the controller

## Conclusion

In this paper, we introduced *WorldLight*, which closes the gap between pixel-based and feature-based TSCs. *WorldLight* has demonstrated that learning the representation of traffic-state images can achieve competitive results compared to manual-designing features. Moreover, *WorldLight* outperforms state-of-the-art methods in some scenarios. The proposed method has shown promise for one intersection. Future work will investigate the performance of this algorithm along a corridor or a larger city network. Furthermore, recent proposed techniques to improve world models in robotic research such as contrastive learning and data augmentation can be potential directions for *WorldLight*.

## References


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

Bishop, C. M. 1994. Mixture density networks.

Chen, C.; Wei, H.; Xu, N.; Zheng, G.; Yang, M.; Xiong, Y.; Xu, K.; and Zhenhui. 2020. Toward a thousand lights: Decentralized deep reinforcement learning for large-scale traffic signal control. In *AAAI*.

David Schrank, Luke Albert, B. E. T. L. 2021. *2021 Urban Mobility Report*. Texas AM Transportation Institute.

Egea, A. C., and Connaughton, C. 2021. Assessment of reward functions in reinforcement learning for multi-modal urban traffic control under real-world limitations*. *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)* 2095–2102.

Ha, D., and Schmidhuber, J. 2018a. Recurrent world models facilitate policy evolution. In Bengio, S.; Wallach, H.; Larochelle, H.; Grauman, K.; Cesa-Bianchi, N.; and Garnett, R., eds., *Advances in Neural Information Processing Systems*.

Ha, D. R., and Schmidhuber, J. 2018b. World models. *ArXiv* abs/1803.10122.

Harris, A.; Stovall, J.; and Sartipi, M. 2019. Mlk smart corridor: An urban testbed for smart city applications. In *2019 IEEE International Conference on Big Data (Big Data)*, 3506–3511.

Hochreiter, S., and Schmidhuber, J. 1997. Long short-term memory. *Neural Comput.* 9(8):1735–1780.

James Bonneson, Srinivasa R Sunkari, M. P. P. S. 2011. *Traffic Signal Operations Handbook*. Texas Department of Transportation.

Liang, X.; Du, X.; Wang, G.; and Han, Z. 2019. A deep reinforcement learning network for traffic light cycle control. *IEEE Transactions on Vehicular Technology* 68(2):1243–1253.

Mei, H.; Lei, X.; Da, L.; Shi, B.; and Wei, H. 2022. Libsignal: An open library for traffic signal control. *Reinforcement Learning for Real Life Workshop at NeurIPS*.

Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A. A.; Veness, J.; Bellemare, M. G.; Graves, A.; Riedmiller, M. A.; Fidjeland, A.; Ostrovski, G.; Petersen, S.; Beattie, C.; Sadik, A.; Antonoglou, I.; King, H.; Kumaran, D.; Wierstra, D.; Legg, S.; and Hassabis, D. 2015. Human-level control through deep reinforcement learning. *Nature* 518:529–533.

Mnih, V.; Badia, A. P.; Mirza, M.; Graves, A.; Harley, T.; Lillicrap, T. P.; Silver, D.; and Kavukcuoglu, K. 2016. Asynchronous methods for deep reinforcement learning. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, ICML'16, 1928–1937. JMLR.org.

Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; and Klimov, O. 2017. Proximal policy optimization algorithms. *ArXiv* abs/1707.06347.

Tran, T. V.; Doan, T.-N.; and Sartipi, M. 2021. Tslib: A unified traffic signal control framework using deep reinforcement learning and benchmarking. In *2021 IEEE International Conference on Big Data (Big Data)*, 1739–1747.

van der Pol, E., and Oliehoek, F. A. 2016. Coordinated deep reinforcement learners for traffic light control.

Varaiya, P. 2013. Max pressure control of a network of signalized intersections. *Transportation Research Part C: Emerging Technologies* 36:177–195.

Wei, H.; Zheng, G.; Yao, H.; and Li, Z. 2018. Intellilight: A reinforcement learning approach for intelligent traffic light control. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery Data Mining*, KDD '18, 2496–2505. New York, NY, USA: Association for Computing Machinery.

Wei, H.; Chen, C.; Zheng, G.; Wu, K.; Gayah, V.; Xu, K.; and Li, Z. 2019. Presslight: Learning max pressure control to coordinate traffic signals in arterial network. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery Data Mining*, KDD '19, 1290–1298. New York, NY, USA: Association for Computing Machinery.

Wei, H.; Zheng, G.; Gayah, V.; and Li, Z. 2021. Recent advances in reinforcement learning for traffic signal control: A survey of models and evaluation. *SIGKDD Explor. Newsl.* 22(2):12–18.

Zheng, G.; Zang, X.; Xu, N.; Wei, H.; Yu, Z.; Gayah, V.; Xu, K.; and Li, Z. 2019. Diagnosing reinforcement learning for traffic signal control.