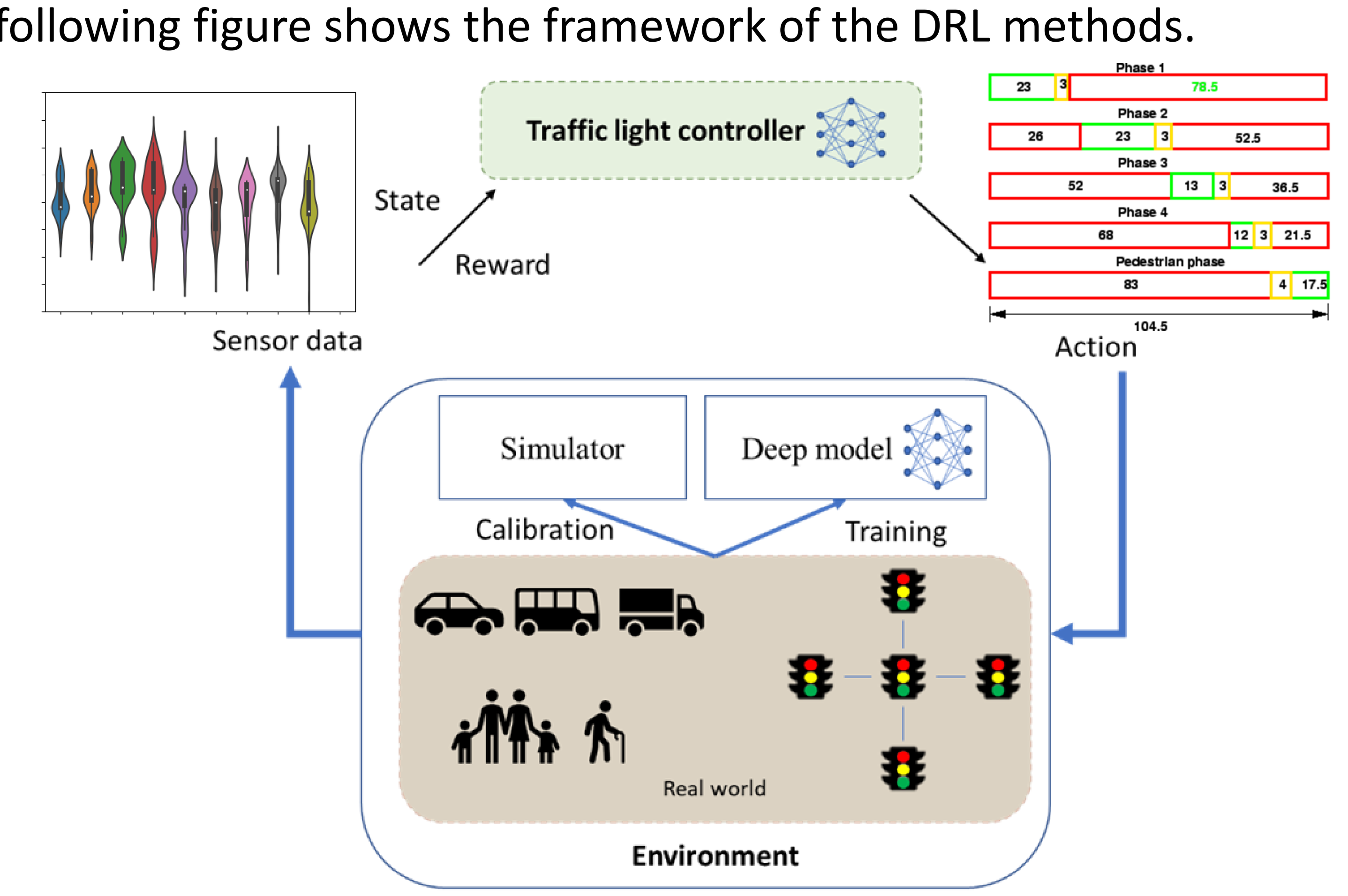
A Model-based Deep Q-Learning Algorithm for Traffic Signal Timing Control at Isolated Intersections

Yun Yuan, Xianfeng Terry Yang* (x.yang@utah.edu), Hao Wang, Tian Zhao, Yang Liu
Department of Civil and Environmental Engineering, University of Utah

INTRODUCTION

To contend the considerable externalities of traffic congestion, researchers and engineers have made great efforts to improve traffic-responsive algorithms for Adaptive Traffic Signal Control (ATSC). Deep reinforcement learning (DRL) methods have been tested on simplified traffic light timing problem and show promising potentials in addressing the curse of dimensionality. However, previous studies ignore the limitation of sensors and heavily rely on simulators. The following figure shows the framework of the DRL methods.



The DRL framework for traffic signal timing

Advantages

✓ the effectiveness of RL and DRL methods (deep policy gradient, value estimation, Q-learning) in traffic signal timing in simulated environment

✓ the effectiveness of traffic simulators (such as VISSIM, SUMO) in developing RL methods

Disadvantages

× computational cost of the simulation sampling

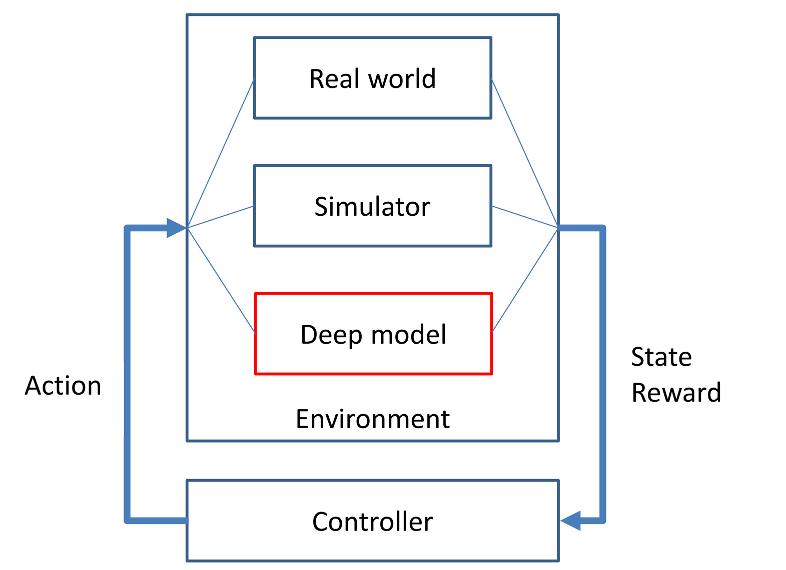
× heavily rely on simulators, which cannot provide plausible results despite costly highly-informative data and fine-tuned parameters

× not based on real data or experiments in real-world scenarios and are hard to implement due to simplistic assumptions

This paper customizes a DQL method to optimize traffic signal timings at single intersections, where a Deep Q Network (DQN) is proposed to estimate the value function (i.e., delays) and another Deep Neural Network is used to synthetize the future state. Then the machine learning-based methods are evaluated on a real-world case with Automatic Number-Plate Recognition (ANPR) data.

a) A deep reinforcement learning algorithm is proposed with vehicle delay distribution state presentation and partially observed flawed inputs.
b) A deep model is proposed to speed up the computation.
c) ANPR field data is used for training the deep neural networks.
d) The field SCOOT data is used as the baseline for benchmarking.

METHODS



The compatible, realistic simulator and deep model

Note it takes about 40 hours on average to test on the simulator SUMO while 1.2 hours on the deep model on a workstation with Intel i5-6600K CPU.

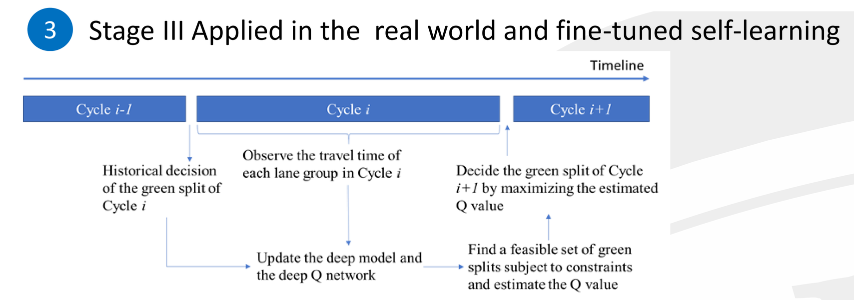
1 Stage I Mimicking expert player

Step 1. For each record of the expert experience, loop Step 2-4;
Step 2. Find a feasible action set A_t subject to Constraints (3)-(5);
Step 3. Generate potential next state $S_{t+1} = G(s_t, A_t)$;
Step 4. Evaluate reward with Equation 6.
Step 5. Update Q with the synthesized $s_t, a_t, r(s_t, a_t), s_{t+1}$.

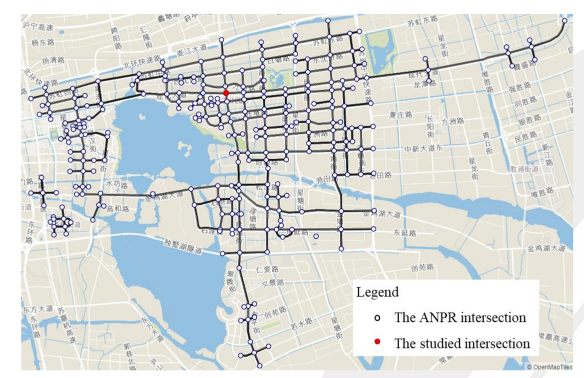
2 Stage II Deep model aided self-learning

Step 1. Initialize s_0, a_0, Q;
Step 2. For $m \in [1, M]$, loop the Steps 3-9;
Step 3. For $t \in [1, T]$, loop the Steps 4-9;
Step 4. Generate new state s_{t+1};
Step 5. Update Q with the generated $s_t, a_t, r(s_t, a_t), s_{t+1}$;
Step 6. Find the feasible action set A_t at the state s_t subject to
Step 7. Generate potential next state $\hat{s}_{t+1}$;
Step 8. Evaluate the Q value of the potential next states $\hat{s}_{t+1}$;
Step 9. Take the action a_t and observe the next state s_{t+1};

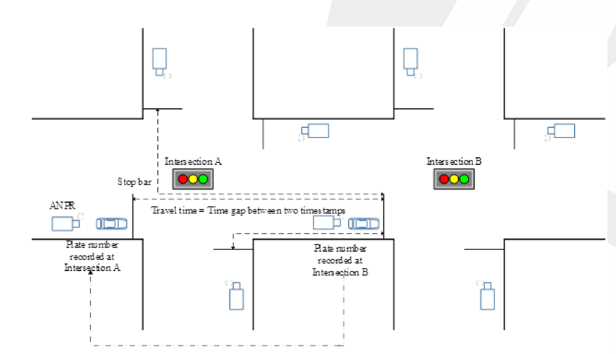
3 Stage III Applied in the real world and fine-tuned self-learning



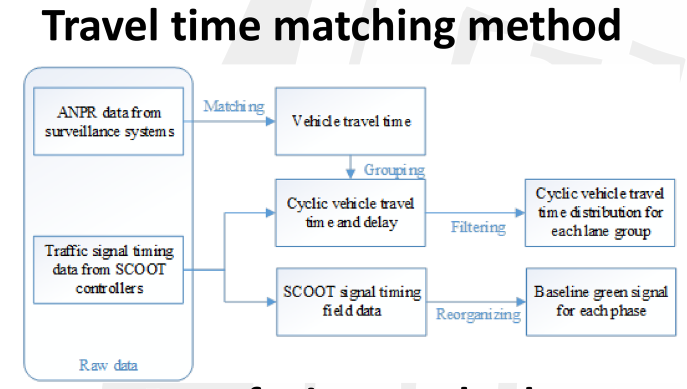
RESULTS/DISCUSSION



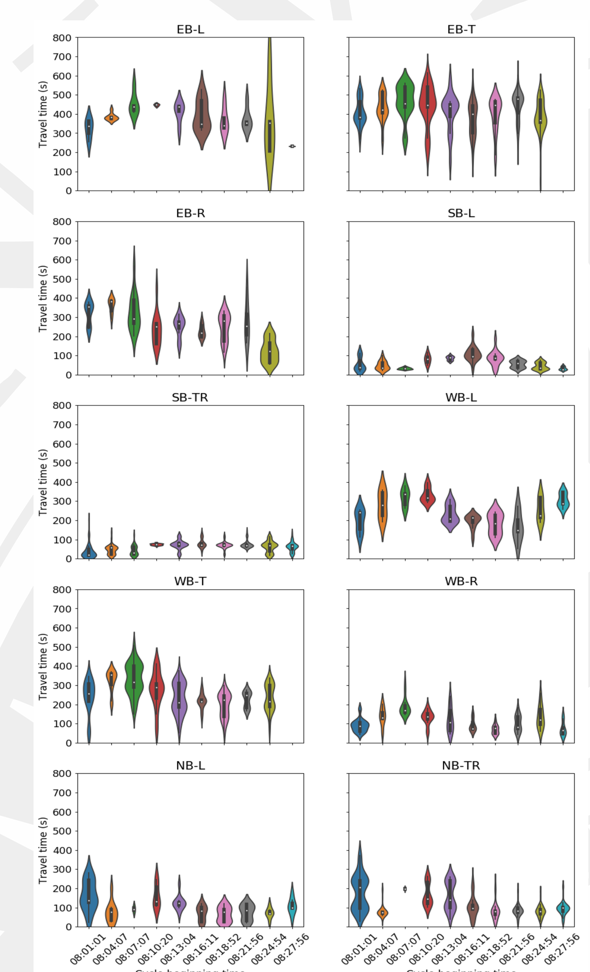
The real world ANPR system network



Travel time matching method

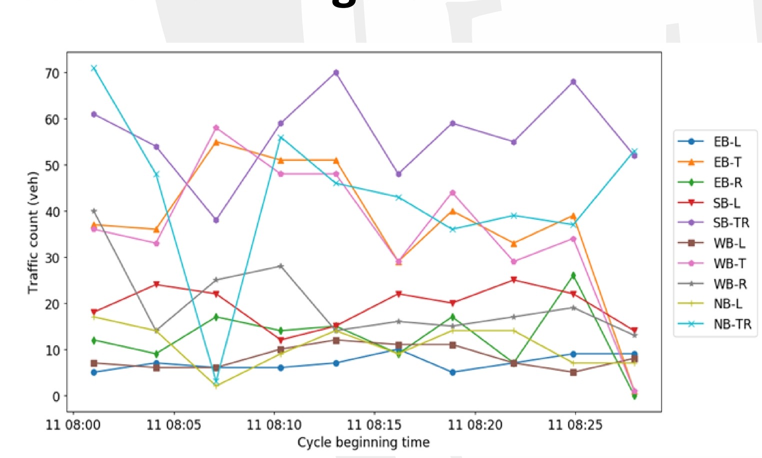


Data fusion method

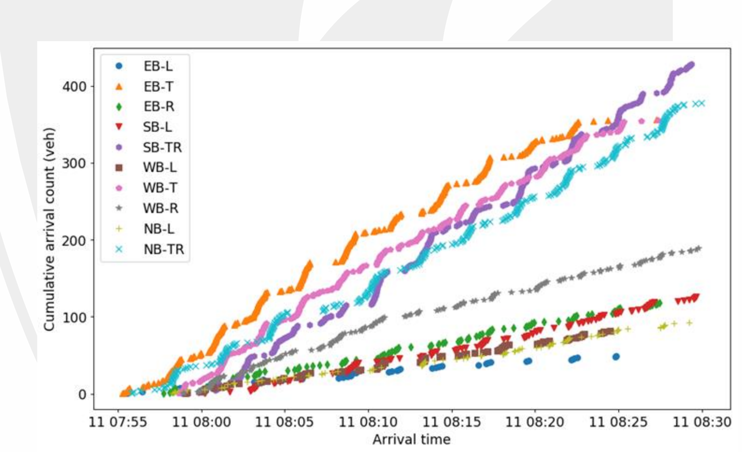


Lane-group travel time distribution violin diagram

Scenario configuration



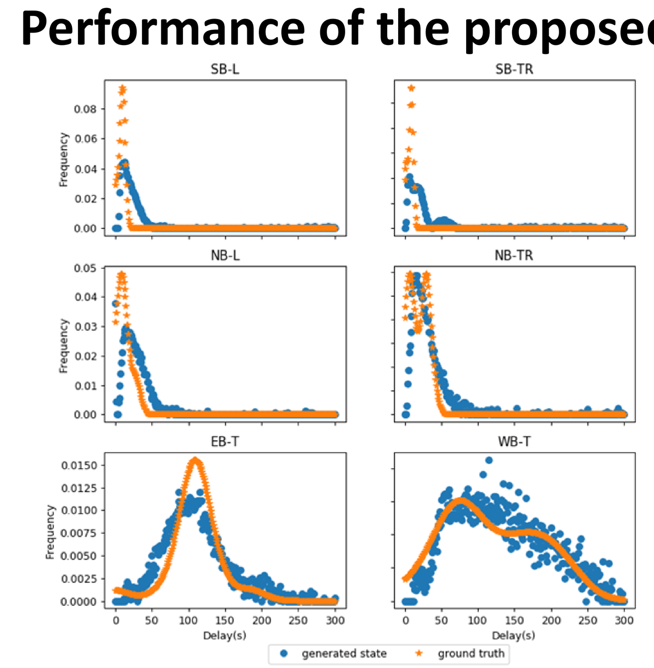
Traffic count pattern



Arrival time pattern

Note in comparison to the aggregated average traffic flow, the disaggregated inputs preserve the realistic vehicle arrival platooning information. In view of the observed overflow queue and the approach spillback, the studied intersection shows oversaturation in the peak hours.

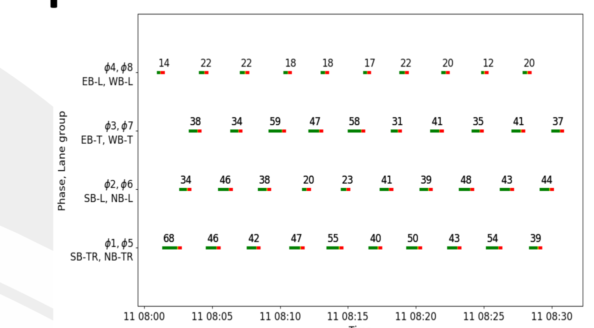
Performance of the proposed deep model



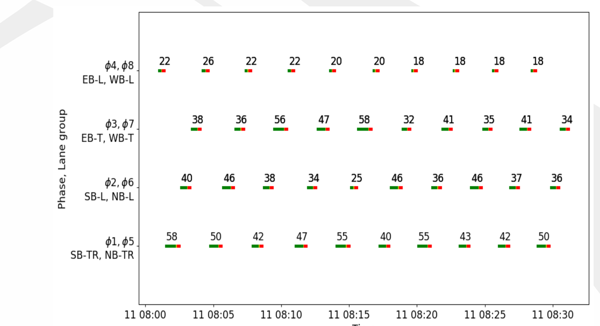
The figure shows the deep model can accurately estimate the delay distribution of 6 lane groups (out of 10 lane groups) at a 4-leg intersection in a morning peak cycle in comparison to the real data, where NB, SB, L, T, and R represent northbound, southbound, left-turn, through, right-turn, respectively. The y-axis is for the probability of that the corresponding the vehicle appears in this cycle. The average R2 and relative mean squared error between the observed and synthesized sample are 0.69 and 0.0072, respectively.

Performance benchmarking of the proposed DRL method

Algorithm	Lane group	Average delay (s)
DQL ★	All	144.47
	SB-L	45.11
	SB-TR	51.42
	NB-L	88.35
	NB-TR	78.88
	EB-T	310.12
	WB-T	188.15
SCOOT	All	150.34
	SB-L	43.62
	SB-TR	44.92
	NB-L	84.42
	NB-TR	102.86
	EB-T	377.58
	WB-T	214.77
SYNCHRO	All	176.25
	SB-L	58.56
	SB-TR	53.34
	NB-L	92.33
	NB-TR	101.2
	EB-T	351.1
	WB-T	200.11



SCOOT operations



The proposed DRL operations

CONCLUSIONS

This paper customizes a Deep Q-learning Learning (DQL) method to optimize traffic signal timings at single selfish intersections and contributes in

a) Designing better state representation (vehicle delay distribution) and reasonable assumptions (compatible across simulators, deep neural networks and real sensors);
b) Developing a deep model to speed up the sampling
c) Testing the proposed methods with data from real-world scenarios (ANPR data);
d) Comparing with commercial systems (SCOOT, SNYCHRO) Experiments show the machine learning-based model can predict the traffic state in limited computational time and the deep Q-learning algorithm is 3.9% better than the field experiment performance from the adaptive control system, SCOOT, and 22% better than the time-of-day plan by SYNCHRO

ACKNOWLEDGEMENTS

- Data source from Suzhou Industrial Park, China
- NVIDIA Corp for GPU donation (4th author)
- NAU-IT Fund RCQD16-01 (5th author)