연세대학교
YONSEI UNIVERSITY

[STA3145] Reinforcement Learning

Adaptive Traffic Signal Control for Seoul Using Multi-Agent Offline RL

Team 10 | Final Project

2020123155 송미형
2022119044 김채원
2025848098 Valentina Wei
2025849380 Kieran Khan

Executive Summary

INTRO

Motivation

- **Limitation of online RL:** expensive, unsafe
- **Extrapolation error in offline RL:** out-of-distribution (ood) actions

Suggestion

- Support-Threshold Fitted) Q-Iteration **(ST-FQI):**
  - ✓ Support gate on Bellman target
  - ✓ Tree-based FQI

ANALYSIS

Model Setup

- **Multi-Agent:** intersection $i \in V$ is treated as one RL agent in a coordination game
- **Queue Dynamics:** $q_i(t+1) = \max\{0,\ q_i(t) + \lambda_i(t)C - s \cdot g_i(t)\}.$
- **State:** $s_i(t) = [q_i(t),\ \lambda_i(t),\ \sum_{j \in N(i)} q_j(t)].$
- **Reward:** $r_i(t)$ is the negative of a queue-area delay proxy

Experiment 1

- **Python-only multi-agent CTDE**
  - ✓ Q-learning > ST-FQI > Classical Baselines
  - ✓ ST-FQI mitigates OOD

Experiment 2

- **SUMO simulation on "J0"**
  - ✓ (Local) ST-FQI > Q-learning > Random Baselines
  - ✓ Robust support threshold

KEY

Contribution

- Reducing the performance gap with online RL with a fixed dataset
- Safe decision with support gate

Future studies

- Evaluate ST-FQI with sparser or more biased datasets
- Enhancing dynamics of SUMO simulation setting

Motivation & Importance

We examine the traffic-signal control problem using RL as an effective tool for transportation-demand management

Increasing Congestion Cost in South Korea

	2018	2019	2020	2021	2022
Traffic Congestion Cost	43.7	45.8	41.1	47.3	48.4
Increasing Rate	12.9	4.8	-10.3	15.1	2.3

(한국교통연구원,「2024 국가 교통정책 평가지표 연구사업-제3권 교통혼잡비용(2022)」) [Trillion KRW, %]

* **Traffic Congestion Cost:** Environmental Pollution Costs + Traffic Accident Costs + Costs from Increased Demand

Seoul incurs the highest traffic congestion in the country, highlighting the need for more effective transport-demand management policies

Motivation & Importance

We bring offline RL's safety to a multi-agent traffic setting with action-support constraints, suggesting **Support-Threshold Fitted Q-Iteration (ST-FQI)** model

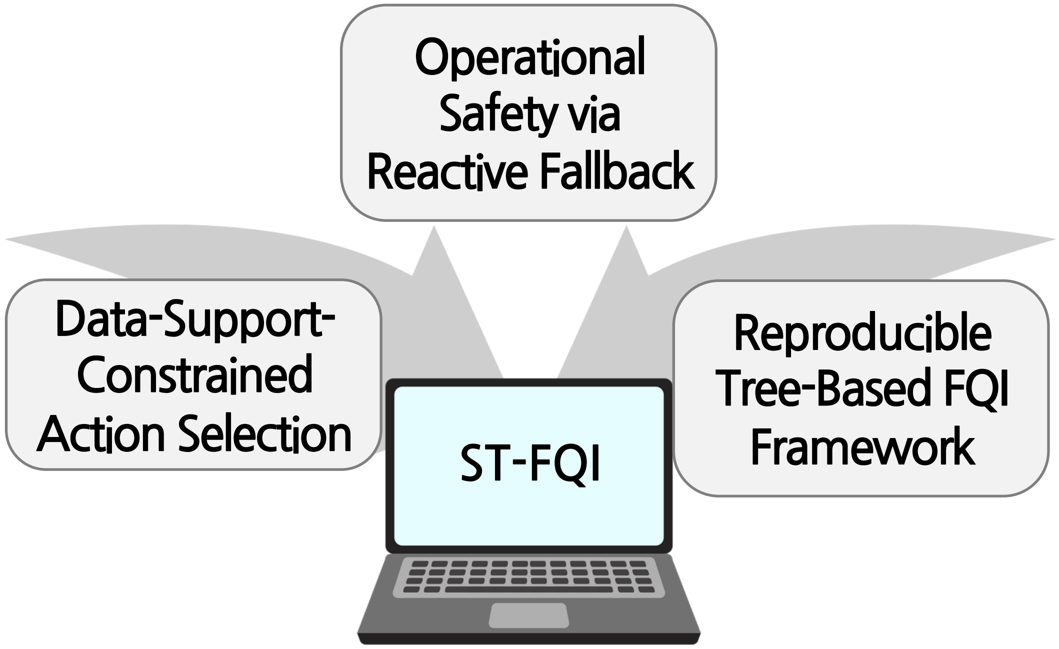
Offline RL

- **Offline RL:** Learning rewarding policy based solely on a dataset of historical interactions (Kidambi et al., 2020)
- The setting is crucial in domains **where active exploration is risky** (e.g. medical treatment, autonomous driving)

Problem: Extrapolation Error

- Function approximators (e.g. deep Q-networks) are forced to extrapolate values for unseen actions, leading to large errors (Kumar et al., 2019)

Our New Suggestion: ST-FQI



Research Question

> We position our work at the intersection of (i) the performance gap between online and offline RL, (ii) offline RL for mitigating extrapolation error, and (iii) RL-based traffic signal control

Stream	Discussion	Related Work
Online vs. Offline RL	• **Online RL:** interactive access to the environment with many deep RL algorithms (e.g., DQN, actor-critic) • **Offline RL:** attractive in safety-critical domains (healthcare, autonomous driving, traffic control)	• Agarwal et al., (2020) • Han et al., (2023)
Offline RL and Extrapolation Error	• Standard off-policy Q-learning updates propagate errors on **out-of-distribution (OOD)** actions • Subsequent iterations cam amplify **bootstrapping error accumulation**	Levine et al., (2020); Kumar et al., (2019); Kumar et al., (2020); Fujimoto et al., (2019); Zhang et al., (2021); Kidambi et al., (2020)
RL for Traffic Signal Control	• RL has been widely adapted to the domain • **Offline RL** in TSC is relatively new, with concern that online RL is impractical in reality due to costs and safety	• Zhang et al., (2023) • Rahman Swapno et al. (2024) • Ming Zhu et al. (2025)

How can we design a safe offline RL that mitigates extrapolation error while reducing the performance gap with online RL in the Traffic Signal Control (TSC) domain?

Problem Setup

We model the traffic network using queue-based dynamics and a multi-agent CTDE framework

Multi Agent CTDE

- **Agents:** Each intersection $i \in V$ is treated as one RL agent in a coordination game
- **State:** $$s_i(t) = [q_i(t),\ \lambda_i(t),\ \sum_{j \in N(i)} q_j(t)].$$
- **Reward:** $r_i(t)$ is the negative of a queue-area delay proxy
- **Queue Dynamics:**

Webster (1958),
Stephanopoulos & Michalopoulos (1979)

$$q_i(t+1) = \max\{0,\ q_i(t) + \lambda_i(t)C - s \cdot g_i(t)\}.$$

- Training uses a CTDE **(Centralized Training, Decentralized Execution)** design

Amato (2024)

Parameters

$q_i(t)$: queue length (vehicles) at the start of the cycle t

$\lambda_i(t)$: arrival rate during cycle t

C: signal cycle length (seconds)

$g_i(t)$: effective green time allocated in cycle t

s: service rate under green

Model Setup

Problem Setup

We model the traffic network using queue-based dynamics and a multi-agent CTDE framework

Multi Agent CTDE

- **Agents:** Each intersection $i \in V$ is treated as one RL agent in a coordination game
- **State:** $$s_i(t) = [q_i(t),\ \lambda_i(t),\ \sum_{j \in N(i)} q_j(t)].$$
- **Reward:** $r_i(t)$ is the negative of a queue area

Use global information during training, but each agent utilizes neighbor messages during execution

(No global controller required)

- Training uses a CTDE **(Centralized Training, Decentralized Execution)** design

Parameters

$q_i(t)$: queue length (vehicles) at the start of the cycle t

$\lambda_i(t)$: arrival rate during cycle t

C: signal cycle length (seconds)

$g_i(t)$: effective green time allocated in cycle t

s: service rate under green

Model Setup

Problem Setup

> We model the traffic network using queue-based dynamics and a multi-agent CTDE framework

Multi Agent CTDE

- **Agents:** Each intersection $i \in V$ is treated as one RL agent in a coordination game
- **State:** $$s_i(t) = [q_i(t),\ \lambda_i(t),\ \sum_{j \in N(i)} q_j(t)].$$
- **Reward:** $r_i(t)$ is the negative of a queue-area delay proxy
- **Queue Dynamics:**

Webster (1958),
Stephanopoulos & Michalopoulos (1979)

$$q_i(t+1) = \max\{0,\ q_i(t) + \lambda_i(t)C - s \cdot g_i(t)\}.$$

Next Queue =
Current Queue + Arrivals - Serviced Vehicles,
(clipped at zero per cycle)

Parameters

$q_i(t)$: queue length (vehicles) at the start of the cycle t

$\lambda_i(t)$: arrival rate during cycle t

C: signal cycle length (seconds)

$g_i(t)$: effective green time allocated in cycle t

s: service rate under green

Model Setup

ST-FQI: Our New Suggestion (See Appendix-A for Pseudo Code)

We newly introduce **Support-Threshold Fitted Q-iteration (ST-FQI)** algorithm, mitigating extrapolation error by integrating a safety gate directly into the Bellman targets

1. Behavioral Dataset	Reactive baseline rollouts with logged tuples $(s_i^k, a_i^k, r_i^k, s_i^{k+1})$
2. Safety Gate via Behavior Support	Learn a behavior classifier $\hat{p}(a \mid s)$
	Gate the Bellman Targets: $\mathcal{A}_\tau(s') = \{a' : \hat{p}(a' \mid s') \geq \tau\}, \quad \hat{y} = r + \gamma \max_{a' \in \mathcal{A}_\tau(s')} Q(s', a').$
	If $\mathcal{A}_\tau(s')$ is empty, **fallback** to the action of highest $\hat{p}(\cdot \mid s')$
3. Fitted Q-Iteration (per agent)	For each agent, fit a tree-ensemble regressor and iterate Bellman Regression using the gated targets above
4. Policy Extraction	Each agent computes Q on allowed actions and applies the one with largest Q $\pi(s) = \arg\max_{a \in \mathcal{A}_\tau(s)} Q(s, a)$

Novelty of ST-FQI

We introduce **Support-Threshold Fitted Q-iteration (ST-FQI)** algorithm, a new offline approach that combines fitted-Q-iteration with statistical support-aware techniques by injecting an action-support gate into the Bellman update

ST-FQI

- Our ST-FQI algorithm sits at the intersection of two threads of prior work
- Statistically motivated way to insert an action-support gate directly into the Bellman backup of FQI

Stream 1. Fitted Q-Iteration and Batch RL

- Applies the Bellman optimality operator and fits a regression model to approximate the Q-function from a fixed dataset of transitions
- Provide stable value-function approximation

Ernst, D. et al., (2005), Munos, R. et al., (2008), Kumar et al., (2019), Fujimoto et al., (2019), Kumar et al., (2020)

Stream 2 - Statistical Clipping & Support-Aware

- Clipping or truncating importance weights is a standard defense
- When the behavior probability is very small, one should either down-weight or avoid that action

Ionides, E. L. (2008), Munos, R., et al. (2016), Laroche, R. et al., (2019)

Data Explanation

We use data published by the Seoul Metropolitan City Government,
enabling our RL model to reflect authentic urban traffic behavior

'01월 서울시 교통량 조사자료 (2025).xlsx'

- Time-of-day traffic volumes for each monitoring point, along with the geographic coordinates

The goal is to preprocess this data into a traffic simulation called SUMO
From there, the RL agents will optimize traffic light timers

Volume Dataset

- Hourly traffic volume data
- Input to the simulation
- We are focusing on 2-3 intersections around Yonsei

Speed Dataset

- Daily average vehicle speed
- This could be used as a validation dataset
    - i.e. what our research is aiming to beat

Empirical Analysis

Data Explanation

We use data published
enabling our RL model

- Time-of-day traffi

c coordinates

The

MO

Volu

set

- Hourly traffic volur
- Input to the simula
- We are focusing on 2-3 intersections around Yonsei

d

lidation dataset

- i.e. what our research is aiming to beat

Data Explanation

We use data published by th
enabling our RL model to ref

).x

- Time-of-day traffic volu ith

SUMO
SIMULATION OF URBAN MOBILITY

An open-source simulator for detailed, vehicle-level urban traffic modeling

The goal is to preprocess this data into a traffic simulation called SUMO
From there, the RL agents will optimize traffic light timers

Volume Dataset

- Hourly traffic volume data
- Input to the simulation
- We are focusing on 2-3 intersections around Yonsei

Speed Dataset

- Daily average vehicle speed
- This could be used as a validation dataset
  - i.e. what our research is aiming to beat

Experiments Outline

We evaluate ST-FQI in two settings, (i) Python-only CTDE environment with queue-based network composed of multiple intersections and (ii) Sumo-based simulation of a single key intersection using real traffic volumes

Evaluation of ST-FQI

Experiment 1. Python-only multi-agent CTDE

Environment

- Multiple intersections with focused on two critical junctions (D-38 and D-42)
- Each intersection observes local queue lengths and limited neighbor information

Baselines

- Fixed-time control (Koonce, P., 2008)
- Responsive control (Hunt, P. B. et al. 1981)
- Online Q-learning

Non-RL (traditional benchmark)

Experiment 2. Sumo-based case study

Environment

- SUMO simulation of part of Gangnam using real traffic volumes
- Focus on intersection J0 (single agent), a key node affected by D-38 and D-42

Baselines

- Random Online
- Online Q-learning

Experiments Outline

We evaluate ST-FQI in two settings, (i) Python-only CTDE environment with queue-based network composed of multiple intersections and (ii) Sumo-based simulation of a single key intersection using real traffic volumes

Average Waiting Time (AWT)

- Average of the vehicles waiting time in the intersection within the time T
- vehicle delay

Throughput (TP)

- # of vehicles that successfully leave the segment within the time T
- efficiency of traffic flow

Average Number of Stops (ANS)

- The average of complete stops for each vehicle
- Smoothness and stability of control

Overall Evaluation Score (OES)

$$OES = -\alpha * AWT + \beta * TP - \gamma * ANS \quad (\alpha, \beta, \gamma > 0)$$

Experiment 1. Python-only multi-agent CTDE

> ST-FQI significantly outperforms fixed-time and responsive control but does not surpass online Q-learning, which is consistent with the offline RL literature

Setup

- Hourly traffic volumes from the Seoul open data ("01월 서울시 교통량 조사자료 (2025)") are converted to per-second arrival rates and fed into a queue-based network
- At each cycle t, the queue length for intersection evolves as (see p.n for detail)

$$q_i(t+1) = \max\{0,\ q_i(t) + \lambda_i(t)C - s \cdot g_i(t)\}.$$

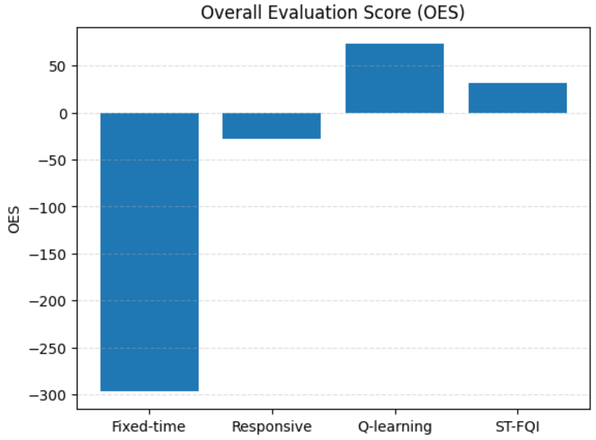
- We compare with 3 policies:

Fixed-time	Based on average demand
Responsive	Smooths recent arrival rates and allocates green proportionally
Q-learning	Tabular Q-learning with ϵ-greedy exploration in the live environment

Overall Evaluation Score (OES)

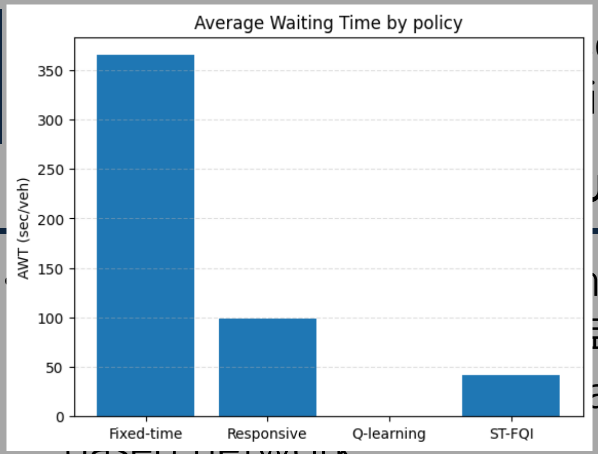
Algorithm	AWT (sec/veh)	TP (veh/episode)	ANS (stops/veh)	OES
Fixed-time	364.96	4575.4	0.904	−297.24
Responsive	99.30	4829.6	0.876	−27.74
Q-learning	**0.03**	4866.4	**0.001**	**72.96**
ST-FQI	41.26	4865.1	0.588	31.13

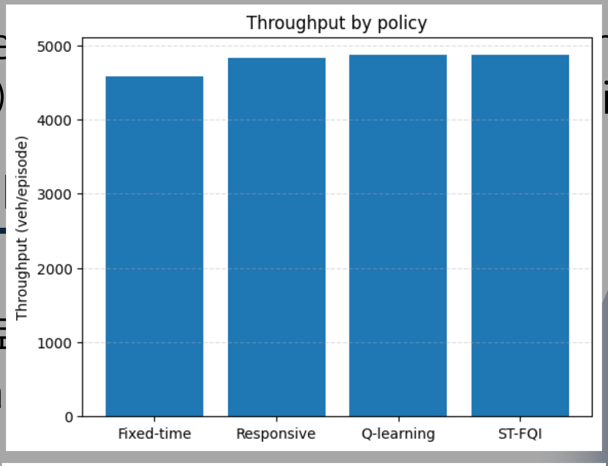
$\tau = 0.05$
$n_{\text{iters}} = 15$

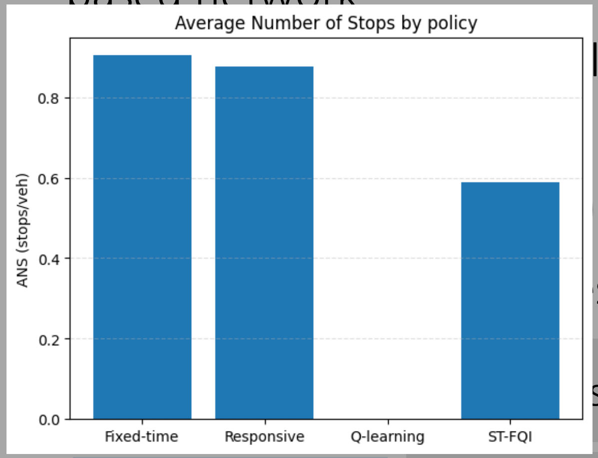


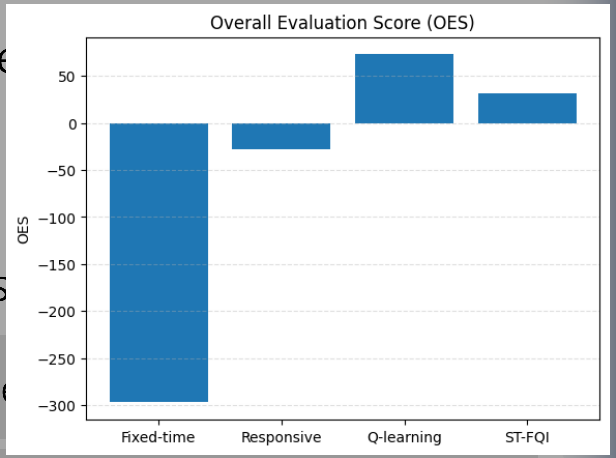
Empirical Analysis

Experiment 1. Python-only multi-agent CTDE





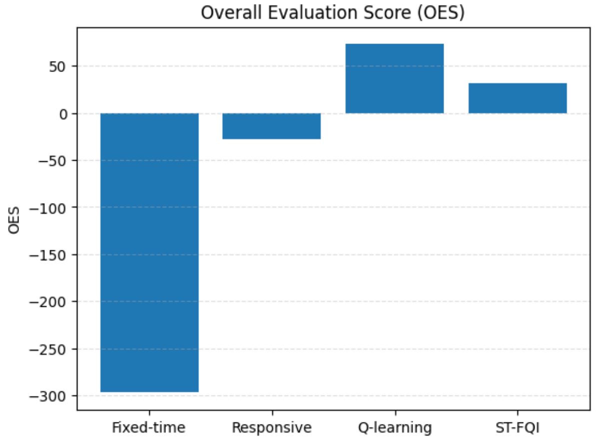




ST-FQI did not beat online Q-learning here, but show offline approach can outperform traditional non-RL benchmarks in TSC which also utilize historical demand data

Overall Evaluation Score (OES)

Algorithm	AWT (sec/veh)	TP (veh/ episode)	ANS (stops/ veh)	OES
Fixed-time	364.96	4575.4	0.904	−297.24
Responsive	99.30	4829.6	0.876	−27.74
Q-learning	**0.03**	4866.4	**0.001**	**72.96**
ST-FQI	41.26	4865.1	0.588	31.13



Smooths recent arrival rates and

Responsive

Global OES: Q-learning > ST-FQI > Responsive > Fixed-Time

Q-learning

exploration in the live environment

Empirical Analysis

Experiment 1. Python-only multi-agent CTDE

We evaluate ST-FQI in two settings, (i) Python-only CTDE environment with queue-based network composed of multiple intersections and (ii) Sumo-based simulation of a single key intersection using real traffic volumes

Setup

- Hourly traffic volumes from the Seoul open data

One of the limitations of offline RL algorithm is that the performance of offline RL methods highly relies on the accuracy of the training simulator (Han et al., 2023)

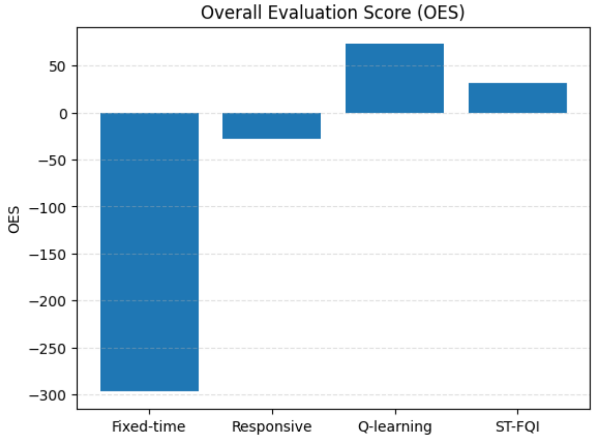
$$q_i(t+1) = \max\{0,\ q_i(t) + \lambda_i(t)C - s \cdot g_i(t)\}.$$

We proceed with Experiment 2 (Sumo simulation) where data-quality issues in Seoul's historical dataset could be alleviated

Q-learning	Tabular Q-learning with ϵ-greedy exploration in the live environment

Overall Evaluation Score (OES)

Algorithm	AWT (sec/veh)	TP (veh/episode)	ANS (stops/veh)	OES
Fixed-time	364.96	4575.4	0.904	−297.24
Responsive	99.30	4829.6	0.876	−27.74
Q-learning	**0.03**	4866.4	**0.001**	**72.96**
ST-FQI	41.26	4865.1	0.588	31.13



Empirical Analysis

Experiment 1. Python-only multi-agent CTDE

ST-FQI constrains Bellman backups to the data support to reduce extrapolation error

Setup

- Hourly traffic volumes from the Seoul open data ("01월 서울시 교통량 조사자료 (2025)") are converted to per-second arrival rates and fed into a queue-based network
- At each cycle t, the queue length for intersection evolves as (see p.6 for detail)

$$q_i(t+1) = \max\{0,\ q_i(t) + \lambda_i(t)C - s \cdot g_i(t)\}.$$

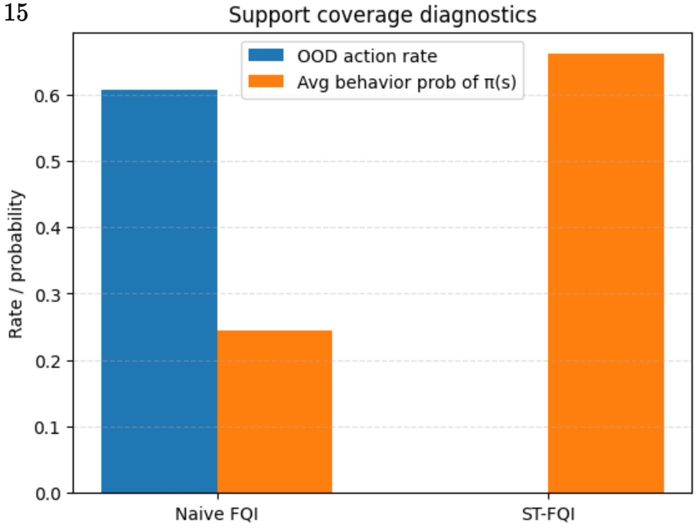
- We compare with 3 policies:

Fixed-time	Based on average demand
Responsive	Smooths recent arrival rates and allocates green proportionally
Q-learning	Tabular Q-learning with ϵ-greedy exploration in the live environment

Probability for Behavior and OOD actions

Algorithm	Avg. behavior prob. Of chosen action	OOD action rate
Naïve FQI	0.244	0.607
ST-FQI	0.661	0.000

$\tau = 0.05$
$n_{\text{iters}} = 15$



Experiment 2. Sumo-based case study

We examine ST-FQI in a more dynamic setting with SUMO simulation, focusing on reducing the performance gap
The chosen configuration (tau = 0.05, n_iters = 40) is numerically best

Setup

- A single signalized intersection (J0) in the Gangnam network using the sumo_rl package (see github attached)
- **State:** (queue length, current phase, neighbor pressure)

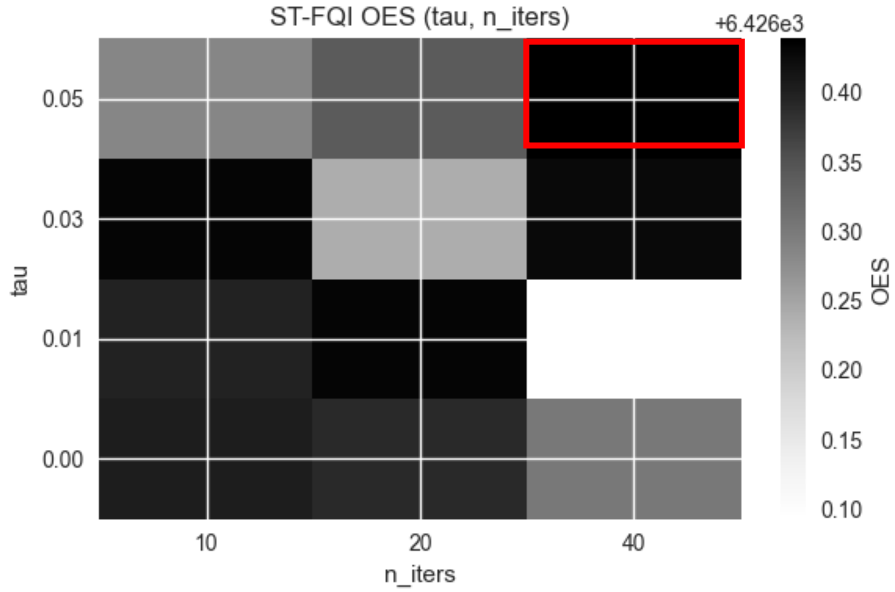
$$s_t = (q_t,\ a_t,\ p_t^{\text{neigh}}) \in \mathbb{R}^3,$$

- **Action:** a discrete phase index where the agent selects signal phase for the next 60-sec cycle
- **Reward**: -(queue length) * cycle length

$$r_t = -q_t \cdot \Delta,$$

Random	Actions are sampled online exactly as during data collection
Q-learning	Tabular / fuction-approximation Q-learning interacting with SUMO (ε-greedy exploration)

Hyper-parameter Tuning for ST-FQI



grid over τ ∈ {0, 0.01, 0.03, 0.05}, n_iters ∈ {10, 20, 40}

Experiment 2. Sumo-based case study

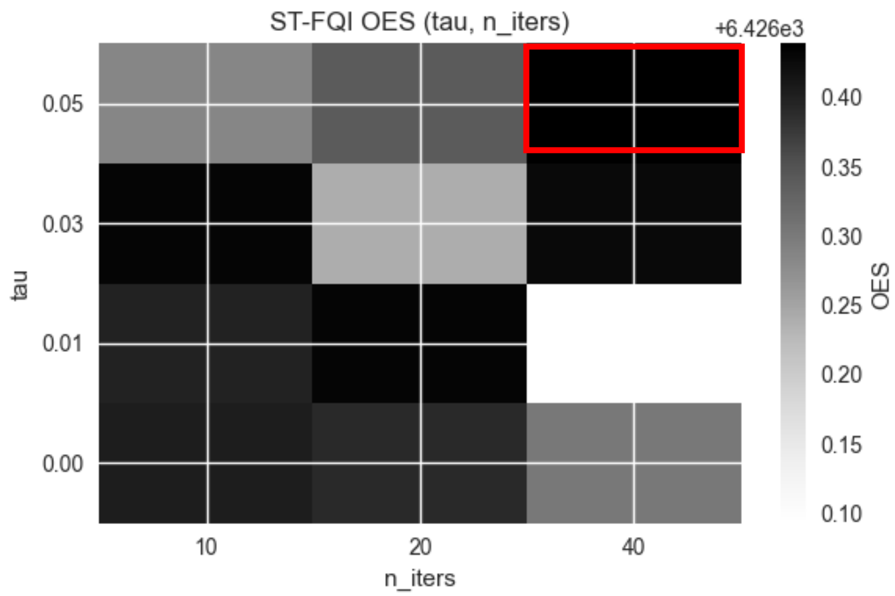
tau	n_iters	label	OES	AWT	TP	ANS
0.00	10	STFQI_tau0p0_it10	6426.403947	0.802400	428481	0.008654
0.00	20	STFQI_tau0p0_it20	6426.390303	0.815887	428481	0.008810
0.00	40	STFQI_tau0p0_it40	6426.301907	0.903986	428481	0.009107
0.01	10	STFQI_tau0p01_it10	6426.397027	0.809233	428481	0.008740
0.01	20	STFQI_tau0p01_it20	6426.430956	0.775379	428481	0.008665
0.01	40	STFQI_tau0p01_it40	6426.092866	1.111961	428481	0.010173
0.03	10	STFQI_tau0p03_it10	6426.432151	0.774268	428481	0.008581
0.03	20	STFQI_tau0p03_it20	6426.239188	0.966822	428481	0.008990
0.03	40	STFQI_tau0p03_it40	6426.427794	0.778578	428481	0.008628
0.05	10	STFQI_tau0p05_it10	6426.285664	0.920022	428481	0.009314
0.05	20	STFQI_tau0p05_it20	6426.339834	0.866267	428481	0.008899
0.05	40	STFQI_tau0p05_it40	6426.438921	0.767465	428481	0.008614

O simulation
merically best

Robustness to Support-Threshold (Tau):
In this SUMO setting, our learned policy is relatively insensitive to the support threshold

(ε-greedy exploration)

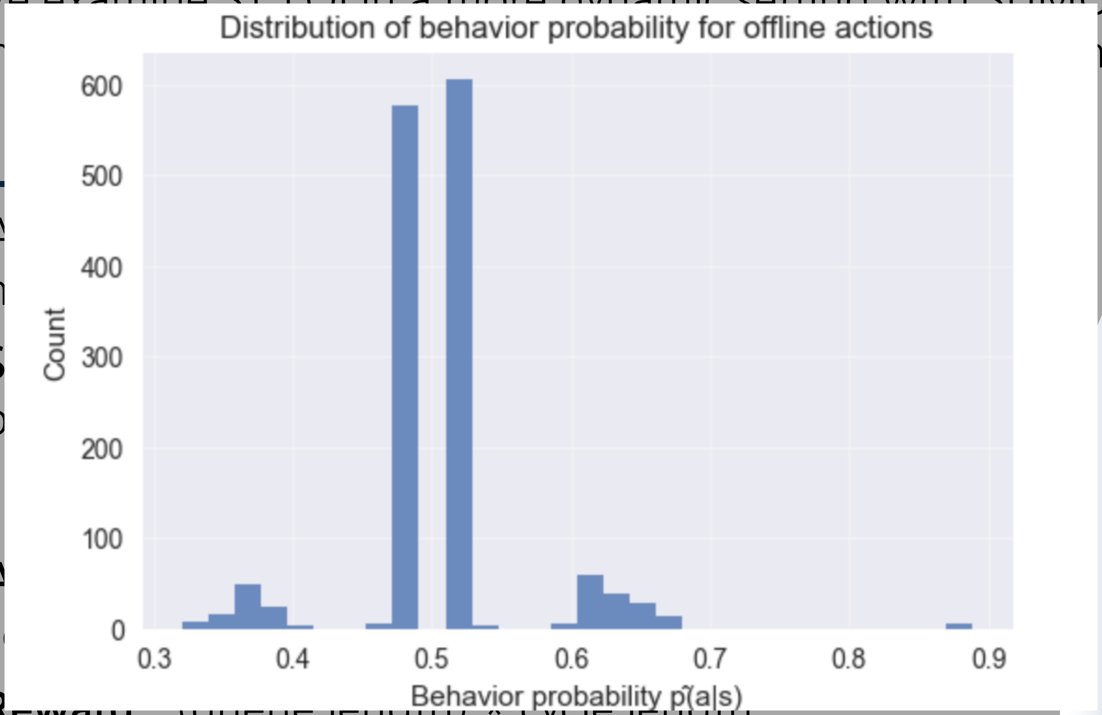
Hyper-parameter Tuning for ST-FQI



grid over τ ∈ {0, 0.01, 0.03, 0.05}, n_iters ∈ {10, 20, 40}

Experiment 2. Sumo-based case study

We examine ST-FQI in a more dynamic setting with SUMO simulation
Th... merically best

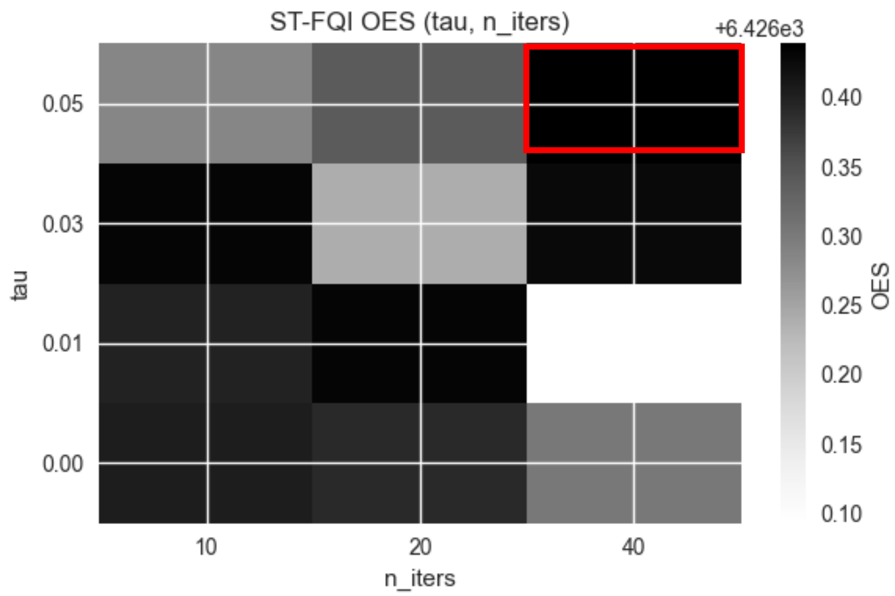


- A ... n ...
- S ... p ...
- A ... s ...
- Reward: (queue length) × cycle length

Enforcing a Support Threshold can be done at little or no performance cost, while providing a safeguard against extrapolation error

Q-learning	tabular / fuction approximation Q learning interacting with SUMO (ε-greedy exploration)

Hyper-parameter Tuning for ST-FQI



grid over $\tau \in \{0, 0.01, 0.03, 0.05\}$, n_iters $\in \{10, 20, 40\}$

Experiment 2. Sumo-based case study

At the targeted intersection J0, ST-FQI edges out Q-learning, especially on average number of stops
The result implies the possibility for that well-designed safe offline RL policy can outperform online RL

Global OES vs. Local OES

	Policy	Global_OES	Local_OES	Global_AWT	Local_AWT	Global_ANS	Local_ANS
0	Random	6426.665956	6427.131216	0.541273	0.082848	0.007772	0.000936
1	Q-learning	6427.025554	6427.178144	0.184757	0.035978	0.004689	0.000878
2	ST-FQI	6426.438921	6427.184264	0.767465	0.030410	0.008614	0.000327

Global Metrics (Whole Network)

- **Q-learning (Best)** > Random > ST-FQI

Local Metrics (Intersection J0 Only)

- **ST-FQI (Best)** > Q-learning > Random

ST-FQI learns a policy that is slightly more self-centered around the offline-controlled intersection J0
Applying ST-FQI to bottle-neck point would lead to better result, aligning with global efficiency

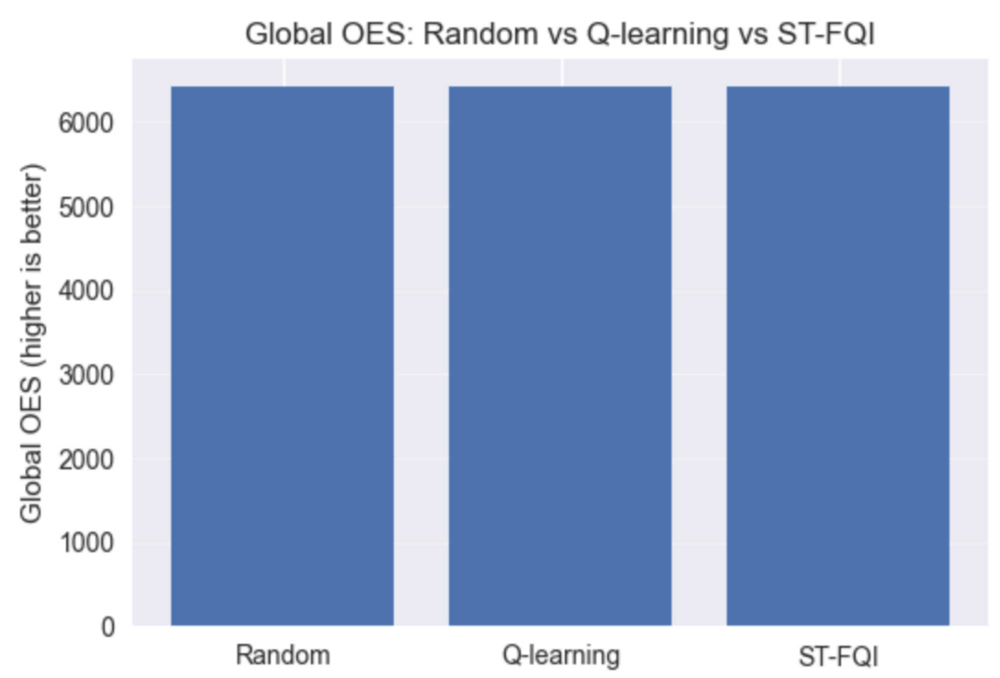
Empirical Analysis

Experiment 2. Sumo-based case study

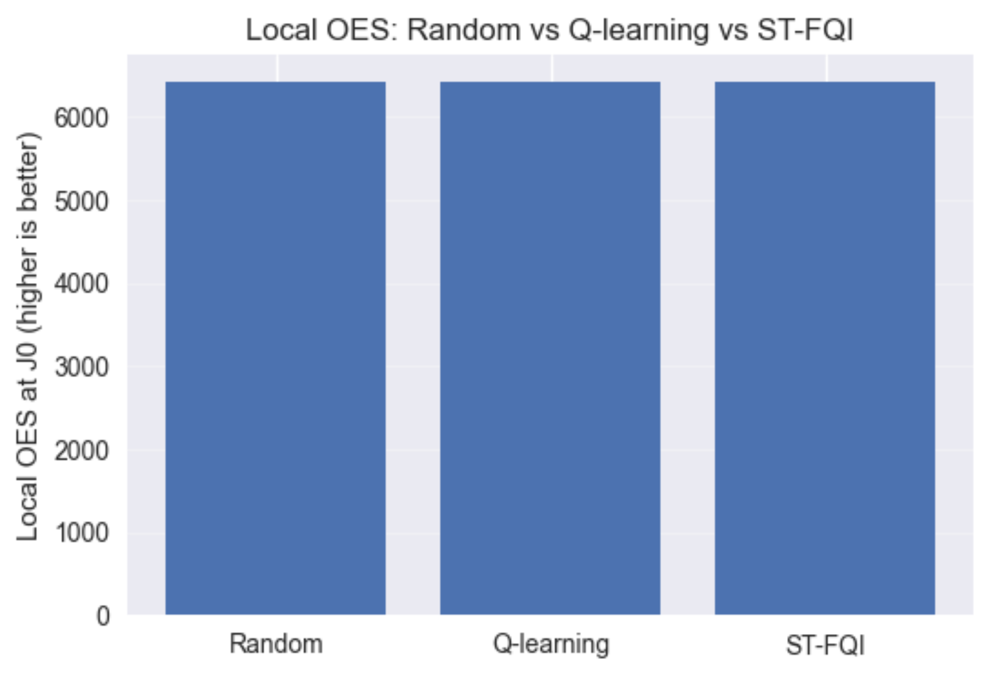
At the ... ning,
The re ... afe c

... vs.

obal_
0.54
0.184
0.767



Global Metrics (Whole Network)



Local Metrics (Intersection J0 Only)

- Q-l

Network-level perspective:
Online Q-learning remains superior, reflecting the known performance gap between online and offline RL

-

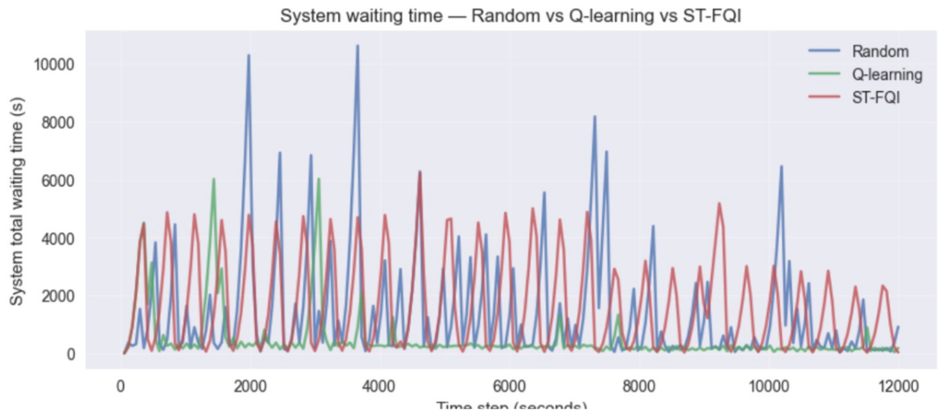
Local Perspective:
ST-FQI closes this gap, indicating offline RL can be competitive with online RL, motivating future studies

ST-FQI learns a policy that is slightly more self-centered around the offline-controlled intersection J0
Applying ST-FQI to bottle-neck point would lead to better result, aligning with global efficiency

Experiment 2. Sumo-based case study

> ST-FQI yields stable traffic flows with a few spikes even with a limited amount of data

Time Series for System Waiting Time



Implications

Efficiency and Stability of ST-FQI

- Even limited offline data can reveal valuable traffic dynamics
- By leveraging a limited amount of data, ST-FQI learns stable policy with better long-term traffic flow

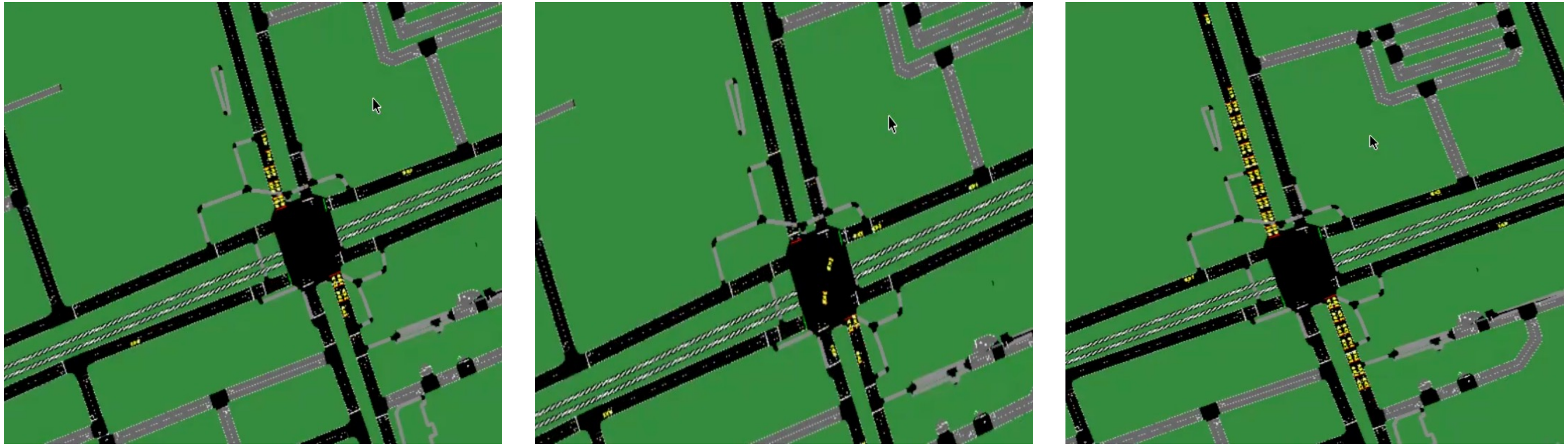
The significance of study for Offline RL

- Online Q-learning benefits from active interaction with SUMO, while ST-FQI is restricted to the logged data
- In real world tasks, online interaction is known to be expensive, slow and unsafe

Experiment 2. Sumo-based case study

Simulation Video: https://drive.google.com/file/d/1ALuKos8OATiLOZB_hoixx9_0_CuJW3bw/view?usp=sharing

```
Warning: Vehicle 'D-38_Inbound_20250101_00.83' performs emergency stop at the end of lane '908696520#6_0' because of a red traffic light (decel=-30.33, offset=5.17), time=1322.00.
Warning: Vehicle 'D-38_Outbound_20250101_00.102' performs emergency braking on lane ':8555608443_0_1' with decel=9.00, wished=4.50, severity=1.00, time=1382.00.
Warning: Vehicle 'D-38_Outbound_20250101_00.102' performs emergency stop at the end of lane '443035112#6_1' because of a red traffic light (decel=-12.04, offset=2.59), time=1382.00.
```



We executed ST-FQI in the SUMO simulator and visualized its real-time traffic dynamics using SUMO GUI

Main Empirical Findings

Our finding is aligned with recent work on conservative offline RL,
which argues that offline approaches are often the right tool when direct online interaction is expensive or risky

Experiment 1. Python-only multi-agent CTDE

Performance

- Substantially reduces AWT and ANS compared to Fixed-time and Responsive policies
- Online Q-learning still has the best OES, but the gap between online Q-learning and offline ST-FQI is much smaller compared to classical baselines

Statistical Implications

- Naïve FQI exhibits higher OOD action rates and low behavior probabilities than ST-FQI

Experiment 2. Sumo-based case study

Performance

- Locally at J0, ST-FQI slightly outperforms Q-learning and random policies
- Motivating future studies to balance global efficiency and local performance of ST-FQI with extension to multi-agent also in SUMO setting

Statistical Implications

- OES is less sensitive to tau, implying safety gate can be executed without sacrificing performance

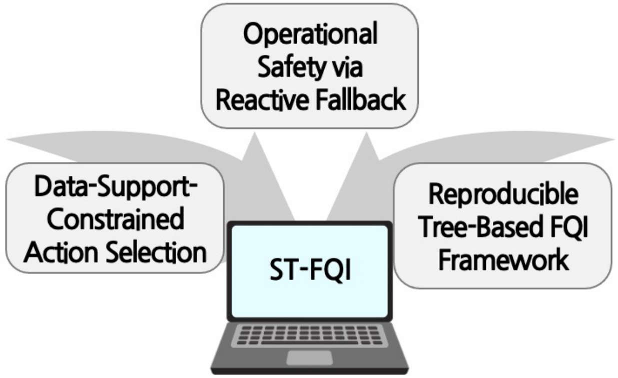
The project illustrates a practical path toward safe, data-driven traffic signal control, while avoiding the instability and operational costs of live exploration

Contribution & Limitation

Despite some limitations, our study reinforces a key message **when online exploration is costly or unsafe,** support-aware offline RL is a compelling alternative capable of **approaching online performance with safety**

Contributions

ST-FQI



Reducing **online-offline RL gap** with a fixed dataset

Avoids **OOD action** and increases the behavior probability of the selected actions

Limitations and Future Work

Limitations

Offline Dataset Quality

Single Agent SUMO

Test ST-FQI under sparser or more biased dataset, with multi agent SUMO simulation

References

Agarwal, R., Schuurmans, D., & Norouzi, M. (2020, November). An optimistic perspective on offline reinforcement learning. In International conference on machine learning (pp. 104-114). PMLR.

Zhang, C., Kuppannagari, S., & Viktor, P. (2021, November). Brac+: Improved behavior regularized actor critic for offline reinforcement learning. In Asian Conference on Machine Learning (pp. 204-219). PMLR.

Kidambi, R., Rajeswaran, A., Netrapalli, P., & Joachims, T. (2020). Morel: Model-based offline reinforcement learning. Advances in neural information processing systems, 33, 21810-21823.

Munos, R., & Szepesvári, C. (2008). Finite-Time Bounds for Fitted Value Iteration. Journal of Machine Learning Research, 9(5).

Ernst, D., Geurts, P., & Wehenkel, L. (2005). Tree-based batch mode reinforcement learning. Journal of Machine Learning Research, 6.

Laroche, R., Trichelair, P., & Des Combes, R. T. (2019, May). Safe policy improvement with baseline bootstrapping. In International conference on machine learning (pp. 3652-3661). PMLR.

Fujimoto, S., Meger, D., & Precup, D. (2019, May). Off-policy deep reinforcement learning without exploration. In International conference on machine learning (pp. 2052-2062). PMLR.

Kidambi, R., Rajeswaran, A., Netrapalli, P., & Joachims, T. (2020). Morel: Model-based offline reinforcement learning. *Advances in neural information processing systems*, *33*, 21810-21823.

Levine, S., Kumar, A., Tucker, G., & Fu, J. (2020). Offline reinforcement learning: Tutorial, review, and perspectives on open problems. arXiv preprint arXiv:2005.01643.

Kumar, A., Fu, J., Soh, M., Tucker, G., & Levine, S. (2019). Stabilizing off-policy q-learning via bootstrapping error reduction. *Advances in neural information processing systems*, *32*.

References

Kumar, A., Zhou, A., Tucker, G., & Levine, S. (2020). Conservative q-learning for offline reinforcement learning. Advances in neural information processing systems, 33, 1179-1191.

Koonce, P. (2008). *Traffic signal timing manual* (No. FHWA-HOP-08-024). United States. Federal Highway Administration.

Munos, R., Stepleton, T., Harutyunyan, A., & Bellemare, M. (2016). Safe and efficient off-policy reinforcement learning. Advances in neural information processing systems, 29.

Han, Y., Wang, M., & Leclercq, L. (2023). Leveraging reinforcement learning for dynamic traffic control: A survey and challenges for field implementation. Communications in Transportation Research, 3, 100104.

Hunt, P. B., Robertson, D. I., Bretherton, R. D., & Winton, R. I. (1981). *SCOOT-a traffic responsive method of coordinating signals* (No. LR 1014 Monograph).

Webster, F. V. (1958). *Traffic signal settings* (No. 39)

Stephanopoulos, G., Michalopoulos, P. G., & Stephanopoulos, G. (1979). Modelling and analysis of traffic queue dynamics at signalized intersections. *Transportation Research Part A: General*, *13*(5), 295-307.

Amato, C. (2024). An introduction to centralized training for decentralized execution in cooperative multi-agent reinforcement learning. *arXiv preprint arXiv:2409.03052*.

Ionides, E. L. (2008). Truncated importance sampling. Journal of Computational and Graphical Statistics, 17(2), 295-311.

Zhang, L., Zhang, Y., Deng, J., & Li, C. (2023). DataLight: Offline Data-Driven Traffic Signal Control. *arXiv preprint arXiv:2303.10828*.

References

Liu, X. Y., Zhu, M., Borst, S., & Walid, A. (2023). Deep reinforcement learning for traffic light control in intelligent transportation systems. *arXiv preprint arXiv:2302.03669*.

Swapno, S. M. R., Nobel, S. N., Islam, M. B., Haque, R., & Rahman, M. M. (2024, February). Traffic light control using reinforcement learning. In *2024 international conference on integrated circuits and communication systems (ICICACS)* (pp. 1-7). IEEE.

Appendix - A

Pseudo Code for ST-FQI

Multi-Agent RL ST-FQI

```
INPUTS
 • Agents: intersections i ∈ V
 • Discrete action set A (green-split adjustments within operational bounds)
 • Pooled offline logs D = ∪i Di, where each Di has tuples (s, a, r, s')
  - s includes local queue q, recent arrivals λ, and a compact neighbor message
 • Regressor class for Q (e.g., tree-ensemble); discount γ
 • Behavior classifier model class for p̂(a | s)
OUTPUTS
 • Per-agent policy πi(s): decentralized controller

PROCEDURE OFFLINE_TRAIN_STFQI (CTDE)
 1) Build/collect offline dataset
   For each agent i:
    – Aggregate logged tuples (s, a, r, s') generated by a safe behavior policy.
   End
   // Training is centralized over pooled data; execution will be decentralized.   (CTDE)

 2) Fit behavior-support model
   Fit a classifier p̂(a | s) on D to approximate the behavior policy.
   Define a supported-action operator:
    Support(s) = { a ∈ A : p̂(a | s) ≥ τ }       // τ is a gating threshold (no value specified)

 3) Initialize action–value functions
   For each agent i:
    Initialize Qi(s, a)  // separate critic per agent; features include the neighbor message
   End

 4) Fitted Q-Iteration with support-aware targets
   Repeat until convergence:
    For each agent i:
     Construct a regression set Ri = ∅
     For each tuple (s, a, r, s') ∈ Di:
      S' = Support(s')                        // compute allowed actions at the next state
      If S' ≠ ∅:
       y = r + γ · max_{a' ∈ S'} Qi(s', a')        // gated Bellman target
      Else:
       a_b = argmax_{a' ∈ A} p̂(a' | s')          // rare fallback to most-supported action
       y = r + γ · Qi(s', a_b)
      Append (s, a, y) to Ri
     End
     Fit/Update Qi by supervised regression on Ri       // tree-ensemble fits for auditability
    End
   End

 5) Policy extraction (decentralized form)
   For each agent i:
    Define πi(s):
     S = Support(s)
     If S ≠ ∅:  πi(s) = argmax_{a ∈ S} Qi(s, a)
     Else:     πi(s) = argmax_{a ∈ A} p̂(a | s)       // behavior fallback
   End

RETURN {πi}_i

PROCEDURE RUN-TIME EXECUTION (Decentralized)
 At each cycle t and for each agent i (in parallel):
  Observe local state s_i(t) (includes neighbor message).
  Choose action a_i(t) = πi(s_i(t)) and clamp to operational bounds if needed.
  Apply the green-split; no centralized coordinator is required.
```

Previous Studies on RL in Traffic Control Problem

Our work is closest to RL-based traffic signal control, but we specifically address extrapolation error using dynamic simulations and strong, realistic baselines

Paper	Approach and Results	Limitations
DataLight: Offline Data-Driven Traffic Signal Control by Liang Zhang et al. (2023)	Use of offline agent and sequential modeling of the state; better results than other offline models.	**Extrapolation error** and limited coordination across intersections
Deep Reinforcement Learning for Traffic Light Control in Intelligent Transportation Systems by Ming Zhu et al. (2025)	Use of DQN for single intersection and DDPG for a grid network; emergence of "greenwave" policy	Highly **theoretical** and very **limited application scenarios**
A Reinforcement Learning Approach for Reducing Traffic Congestion using Deep Q-Learning by Rahman Swapno et al. (2024)	Ise of DQL agent and hyperparameter optimization; 49% queue reduction	**Limited baseline** comparison (only comparing DQL training vs testing)

Our Contribution

Mitigating Extrapolation

- Train a behavior classifier and safety gate to avoid Out-of-Distribution actions

Dynamic Simulation with SUMO

- SUMO simulator captures real-world traffic dynamics

Reasonable Baseline

- Fixed-time, Responsive
- Online Q-learning
- Random policy

DataLight: Offline Data-Driven Traffic Signal Control by Liang Zhang et al. (2023)

Offline RL approach: train control policy from pre-collected (logged) data rather than relying on continuous online interaction.

Sequential modeling of the state: the model captures vehicular speed information within the environment and it then segments roads to capture spatial information, which is further enhanced with sequential modeling it is show that this method outperforms other online/offline traffic signal control methods.

Outline
The paper addresses deployment concerns (offline setting, realism of state representations, real world data); practical inspiration for state design, reward design and offline-batch approach. The model outperforms all other offline models when trained on the real-world dataset COD.

Issue 1

Extrapolation error: still reliant on the completeness of the logged data. If the dataset lacks diversity then offline RL can struggle.

Issue 2

Limited Coordination Across Intersections: DataLight models intersections independently rather than fully modeling a joint multi-intersection policy

Deep Reinforcement Learning for Traffic Light Control in Intelligent Transportation Systems by Ming Zhu et al. (2025)

Deep Q-Network (DQN) for a single intersection: it delivers a thresholding policy for this smaller-scale case.

Deep Deterministic Policy Gradient (DDPG) for a grid network: has the capability to produce on its own a high-level intelligent behavior (i.e. the "greenwave" policy emerges).

Outline

The appendix rigorously proves that under symmetric traffic flow assumptions, the "greenwave" is the unique optimal policy minimizing long-term congestion cost.
For grids, DDPG learns this coordinated pattern without being explicitly programmed for coordination, showing potential for self-organizing control.

Issue 1

Highly theoretical: the static and homogeneous assumptions, simplified traffic environment and experimentation on a 5x10 grid greatly impact real-world implementation

Issue 2

Single-Agent DDP on a grid: the DDPG controller treats all intersections jointly as one large agent, which is computationally expensive and lacks decentralized or multi-agent scalability.

Appendix - B.4

A Reinforcement Learning Approach for Reducing Traffic Congestion using Deep Q-Learning by Rahman Swapno et al. (2024)

Deep Q-Learning (DQL): agent that dynamically manages signal phases at a 4-way intersection using 2 XML datasets of vehicle and route information.

Hyperparameter optimization: stronger than many earlier DQN-based TSC works.

Novel action-selection index: the term for exploration Index(s_i,a) combines action frequency and variance.

Outline

The paper achieved a 49% queue reduction and 9% reward increase. The transparent experimental design and involvement in the Intelligent Transportation Systems development allow for its potential implementation and application.

Issue 1

Limited Baseline Comparison: the experiments compare only DQL's training vs. testing, without quantitative benchmarking against fixed-time, max-pressure, or other RL baselines.

Issue 2

Overfitting Risk and Validation Gap: training and testing are run on the same intersection topology; cross-validation on unseen geometries or demand patterns is absent.

Dataset Description (‘01월 서울시 교통량 조사자료(2025)’)

	A	B	C	D
1	**지점별 일자별 교통량 범례**			
2	구분	설명	표현 예시	예시 설명
3	일자	교통량 조사 일자	20181201	43435
4	요일	교통량 조사 요일 (※ 공휴일은 '일'로 표시)	토	토요일
5	지점명	교통량 조사 도로명(조사지점명)	성산로(금화터널)	조사지점의 도로명(지점명)
6	지점번호	조사지점을 5개 영역(A,B,C,D,F)으로 구분하고 일련번호를 부여함 - [A(도심), B(시계), C(교량), D(간선도로), F(도시고속도로)]	A-01	도심 1번 지점
7	방향	유입 : 외곽에서 서울시청으로 들어오는 방향 유출 : 시울시청에서 외곽으로 나가는 방향	유입/유출	
8	구분	조사지점에서 가까운 교차로명으로 방향표시	봉원고가차도→독립문역	봉원고가차도에서 독립문역 방향의 교통량
9	시간대	1시간 단위를 표시	0시	0시~1시
10	교통량	1시간 교통량	809	809대/시

Appendix - C.2

Dataset Description ('01월 서울시 교통량 조사자료(2025)')

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA
1	일자	요	지점명	지점번	방향	구분	0시	1시	2시	3시	4시	5시	6시	7시	8시	9시	10시	11시	12시	13시	14시	15시	16시	17시	18시	19시	20시
2	20250101	일	성산로(금화터널)	A-01	유입	봉원고가차도->독립문역	583	451	289	200	211	348	580	492	664	780	1045	1139	1265	1149	1327	1186	1177	1134	957	832	708
3	20250102	목	성산로(금화터널)	A-01	유입	봉원고가차도->독립문역	253	162	135	136	248	667	1574	2373	2334	1676	1545	1421	1497	1553	1586	1649	1648	1689	1547	1243	996
4	20250103	금	성산로(금화터널)	A-01	유입	봉원고가차도->독립문역	423	312	225	171	290	603	1508	2183	2233	1884	1867	1861	1677	1693	1647	1705	1826	1858	1537	1285	1073
5	20250104	토	성산로(금화터널)	A-01	유입	봉원고가차도->독립문역	506	376	306	259	235	432	823	882	1117	1479	1576	1523	1639	1451	1549	1301	1160	1056	1047	870	915
6	20250105	일	성산로(금화터널)	A-01	유입	봉원고가차도->독립문역	388	268	241	193	160	252	469	551	541	1038	1147	885	1054	1052	1052	1039	977	851	765	706	629
7	20250106	월	성산로(금화터널)	A-01	유입	봉원고가차도->독립문역	226	155	165	143	248	655	1592	2275	2251	1825	1664	1734	1548	1582	1473	1599	1635	1774	1479	1181	1029
8	20250107	화	성산로(금화터널)	A-01	유입	봉원고가차도->독립문역	447	308	190	177	285	637	1585	2279	2318	1950	2000	1919	1678	1642	1681	1720	1770	1858	1551	1227	1026
9	20250108	수	성산로(금화터널)	A-01	유입	봉원고가차도->독립문역	487	323	214	180	320	620	1503	2219	2274	2012	1790	1876	1710	1554	1631	1734	1666	1831	1662	1240	1020
10	20250109	목	성산로(금화터널)	A-01	유입	봉원고가차도->독립문역	494	302	216	183	282	665	1476	2221	2277	1968	1757	1814	1787	1606	1788	1733	1673	1772	1546	1206	1014
11	20250110	금	성산로(금화터널)	A-01	유입	봉원고가차도->독립문역	493	351	261	177	321	601	1427	2255	2295	2056	2002	1914	1885	1752	1739	1930	2024	2072	1731	1396	1087
12	20250111	토	성산로(금화터널)	A-01	유입	봉원고가차도->독립문역	520	375	302	282	265	421	725	855	1179	1591	1840	1912	1764	1758	1630	1444	1247	1210	1102	980	914
13	20250112	일	성산로(금화터널)	A-01	유입	봉원고가차도->독립문역	446	292	235	167	175	280	505	686	997	1345	1570	1550	1552	1657	1547	1445	1302	1214	1129	956	818
14	20250113	월	성산로(금화터널)	A-01	유입	봉원고가차도->독립문역	285	202	161	148	261	653	1570	2276	2277	2046	1794	1757	1591	1660	1571	1661	1673	1712	1563	1124	976
15	20250114	화	성산로(금화터널)	A-01	유입	봉원고가차도->독립문역	452	266	225	151	281	567	1408	1903	2300	1903	1859	1738	1607	1595	1479	1960	1793	1817	1641	1242	1011
16	20250115	수	성산로(금화터널)	A-01	유입	봉원고가차도->독립문역	488	307	241	210	296	622	1386	2081	2295	1862	1893	1914	1817	1647	1695	1789	1754	1902	1564	1321	1003
17	20250116	목	성산로(금화터널)	A-01	유입	봉원고가차도->독립문역	537	327	229	176	263	606	1385	2127	2222	1752	1780	1979	1677	1525	1602	1718	1738	1854	1613	1228	974
18	20250117	금	성산로(금화터널)	A-01	유입	봉원고가차도->독립문역	511	380	291	205	300	582	1319	2165	2324	1968	2015	1983	1746	1822	1891	1911	1943	2033	1753	1423	1165
19	20250118	토	성산로(금화터널)	A-01	유입	봉원고가차도->독립문역	608	405	347	287	282	414	738	957	1298	1705	1868	1834	1801	1736	1756	1840	1562	1460	1401	1299	927
20	20250119	일	성산로(금화터널)	A-01	유입	봉원고가차도->독립문역	422	352	278	213	183	261	555	796	1150	1451	1612	1535	1512	1540	1556	1412	1353	1206	1141	907	811
21	20250120	월	성산로(금화터널)	A-01	유입	봉원고가차도->독립문역	275	186	166	168	261	644	1414	2276	2392	2040	1884	1899	1768	1740	1768	1883	1763	1975	1648	1231	1057
22	20250121	화	성산로(금화터널)	A-01	유입	봉원고가차도->독립문역	502	301	214	189	281	586	1529	2221	2340	1680	1726	1962	1768	1567	1664	1949	1853	1867	1641	1303	1014
23	20250122	수	성산로(금화터널)	A-01	유입	봉원고가차도->독립문역	547	396	232	195	297	607	1482	2175	2355	1971	1857	2037	1776	1858	1733	1917	1894	2149	1748	1308	1160
24	20250123	목	성산로(금화터널)	A-01	유입	봉원고가차도->독립문역	525	365	276	214	273	624	1439	2236	2366	1681	1837	2001	1744	1756	1735	1903	1984	2118	1665	1237	1105
25	20250124	금	성산로(금화터널)	A-01	유입	봉원고가차도->독립문역	505	389	284	195	311	619	1484	2180	2367	2098	1977	1984	1804	1847	1765	1868	1868	2045	1630	1357	1022

Reproducible Code

Link to Github: https://github.com/chewon1227/ST-FQI

Experiment 1. Python-only multi-agent CTDE

- See ‘Experiment 1.ipynb’ on the main page
- Dataset included in ‘data’ folder

Experiment 2. Sumo-based case study

- See ‘Experiment 2.ipynb’ on the main page
- See ‘README’ and complete the setup for Sumo simulation

(1) Offline Data collection: sumo_rl/collect_data.py
(2) Training ST-FQI Agents: sumo_rl/train_fqi.py
(3) Evaluating Policies (Random, Q-learning, ST-FQI): sumo_rl/evaluate_fqi.py

End of Document