SAFE REINFORCEMENT LEARNING FOR INTERSECTION MANAGEMENT IN RITI COMMUNITIES UNDER RARE EXTREME EVENTS

FINAL PROJECT REPORT

by

Yuanzhang Xiao
University of Hawaii at Manoa

for

Center for Safety Equity in Transportation (CSET)
USDOT Tier 1 University Transportation Center
University of Alaska Fairbanks
ELIF Suite 240, 1764 Tanana Drive
Fairbanks, AK 99775-5910

In cooperation with U.S. Department of Transportation,
Research and Innovative Technology Administration (RITA)

CSET
rural ◦ isolated ◦ tribal ◦ indigenous

DISCLAIMER

The contents of this report reflect the views of the authors, who are responsible for the facts and the accuracy of the information presented herein. This document is disseminated under the sponsorship of the U.S. Department of Transportation’s University Transportation Centers Program, in the interest of information exchange. The Center for Safety Equity in Transportation, the U.S. Government and matching sponsor assume no liability for the contents or use thereof.

TECHNICAL REPORT DOCUMENTATION PAGE

1. Report No.

2. Government Accession No.

3. Recipient's Catalog No.

4. Title and Subtitle

Safe Reinforcement Learning for Intersection Management in RITI Communities Under Rare Extreme Events

5. Report Date

November 5, 2024

6. Performing Organization Code

7. Author(s) and Affiliations

Yuanzhang Xiao, University of Hawaii at Manoa

8. Performing Organization Report No.

INE/CSET 24.19

9. Performing Organization Name and Address

Center for Safety Equity in Transportation
ELIF Building Room 240, 1760 Tanana Drive
Fairbanks, AK 99775-5910

10. Work Unit No. (TRAIS)

11. Contract or Grant No.

12. Sponsoring Organization Name and Address

United States Department of Transportation
Research and Innovative Technology Administration
1200 New Jersey Avenue, SE
Washington, DC 20590

13. Type of Report and Period Covered

Final Report

14. Sponsoring Agency Code

15. Supplementary Notes

Report uploaded to:

16. Abstract

The rapid advancement of artificial intelligence (AI) is reshaping the transportation sector, with applications spanning autonomous vehicles, driver injury prevention, and traffic management. Efficient traffic management, particularly through adaptive intersection control, holds significant potential for reducing congestion. This study explores the application of reinforcement learning (RL) to adaptive traffic signal control in rural, isolated, tribal, and indigenous (RITI) communities, which face unique challenges such as rare extreme weather events. Standard RL approaches struggle in these contexts due to limited exposure to these rare events.

In our study, we first evaluate several mainstream RL algorithms and identified two most promising approaches. Then, we propose to use offline RL algorithms, which can train on existing datasets before interacting with the real environments. This provides a robust solution because (1) it is costly to deploy the algorithm and let the traffic network operate under suboptimal policies before the algorithm learns the optimal policy, and (2) it mimics the scenario where some events are not seen in the training dataset. We compare the performance of offline RL algorithms using different offline datasets, generated by policies of different levels of expertise, in realistic test cases. Results indicate that offline RL approaches perform better when trained on datasets from expert policies, stressing the importance of the quality of the offline datasets. These findings highlight the potential of RL-based adaptive traffic control for improving transportation efficiency, especially when tailored to the specific conditions of RITI communities.

17. Key Words

Machine learning, Traffic signal control, Reinforcement learning

18. Distribution Statement

19. Security Classification (of this report)

Unclassified.

20. Security Classification (of this page)

Unclassified.

21. No. of Pages

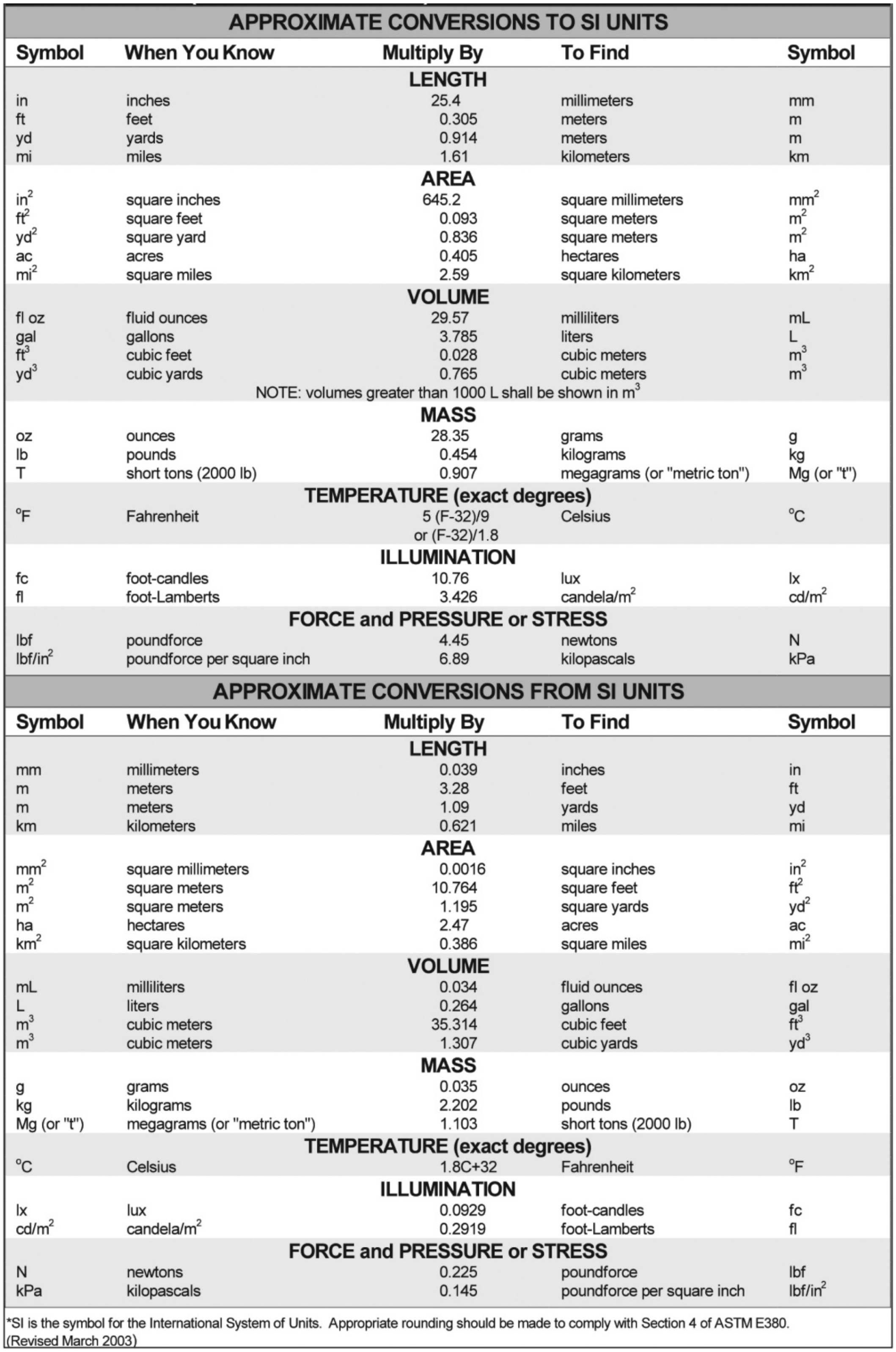
29

22. Price

N/A

Reproduction of completed page authorized.

SI* (MODERN METRIC) CONVERSION FACTORS



APPROXIMATE CONVERSIONS TO SI UNITS

Symbol	When You Know	Multiply By	To Find	Symbol
		LENGTH		
in	inches	25.4	millimeters	mm
ft	feet	0.305	meters	m
yd	yards	0.914	meters	m
mi	miles	1.61	kilometers	km
		AREA		
in^2	square inches	645.2	square millimeters	mm^2
ft^2	square feet	0.093	square meters	m^2
yd^2	square yard	0.836	square meters	m^2
ac	acres	0.405	hectares	ha
mi^2	square miles	2.59	square kilometers	km^2
		VOLUME		
fl oz	fluid ounces	29.57	milliliters	mL
gal	gallons	3.785	liters	L
ft^3	cubic feet	0.028	cubic meters	m^3
yd^3	cubic yards	0.765	cubic meters	m^3
		NOTE: volumes greater than 1000 L shall be shown in m^3		
		MASS		
oz	ounces	28.35	grams	g
lb	pounds	0.454	kilograms	kg
T	short tons (2000 lb)	0.907	megagrams (or "metric ton")	Mg (or "t")
		TEMPERATURE (exact degrees)		
°F	Fahrenheit	5 (F-32)/9 or (F-32)/1.8	Celsius	°C
		ILLUMINATION		
fc	foot-candles	10.76	lux	lx
fl	foot-Lamberts	3.426	candela/m^2	cd/m^2
		FORCE and PRESSURE or STRESS		
lbf	poundforce	4.45	newtons	N
lbf/in^2	poundforce per square inch	6.89	kilopascals	kPa

APPROXIMATE CONVERSIONS FROM SI UNITS

Symbol	When You Know	Multiply By	To Find	Symbol
		LENGTH		
mm	millimeters	0.039	inches	in
m	meters	3.28	feet	ft
m	meters	1.09	yards	yd
km	kilometers	0.621	miles	mi
		AREA		
mm^2	square millimeters	0.0016	square inches	in^2
m^2	square meters	10.764	square feet	ft^2
m^2	square meters	1.195	square yards	yd^2
ha	hectares	2.47	acres	ac
km^2	square kilometers	0.386	square miles	mi^2
		VOLUME		
mL	milliliters	0.034	fluid ounces	fl oz
L	liters	0.264	gallons	gal
m^3	cubic meters	35.314	cubic feet	ft^3
m^3	cubic meters	1.307	cubic yards	yd^3
		MASS		
g	grams	0.035	ounces	oz
kg	kilograms	2.202	pounds	lb
Mg (or "t")	megagrams (or "metric ton")	1.103	short tons (2000 lb)	T
		TEMPERATURE (exact degrees)		
°C	Celsius	1.8C+32	Fahrenheit	°F
		ILLUMINATION		
lx	lux	0.0929	foot-candles	fc
cd/m^2	candela/m^2	0.2919	foot-Lamberts	fl
		FORCE and PRESSURE or STRESS		
N	newtons	0.225	poundforce	lbf
kPa	kilopascals	0.145	poundforce per square inch	lbf/in^2

*SI is the symbol for the International System of Units. Appropriate rounding should be made to comply with Section 4 of ASTM E380.
(Revised March 2003)

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

EXECUTIVE SUMMARY

The rapid advancement of artificial intelligence (AI) is reshaping the transportation sector, with applications spanning autonomous vehicles, driver injury prevention, and traffic management. Efficient traffic management, particularly through adaptive intersection control, holds significant potential for reducing congestion. This study explores the application of reinforcement learning (RL) to adaptive traffic signal control in rural, isolated, tribal, and indigenous (RITI) communities, which face unique challenges such as rare extreme weather events. Standard RL approaches struggle in these contexts due to limited exposure to these rare events.

In our study, we first evaluate several mainstream RL algorithms, including DQN, PPO, QR-DQN, TRPO, ARS, and A2C, in two evaluation cases, and identified DQN and PPO as the most promising approaches. Then, we propose to use offline RL algorithms, which can train on existing datasets before interacting with the real environments. This provides a robust solution because (1) it is costly to deploy the algorithm and let the traffic network operate under suboptimal policies before the algorithm learns the optimal policy, and (2) it mimics the scenario where some events are not seen in the training dataset. We compare the performance of offline RL algorithms using different offline datasets, generated by policies of different levels of expertise, in realistic test cases. Results indicate that offline RL approaches perform better when trained on datasets from expert policies, stressing the importance of the quality of the offline datasets. These findings highlight the potential of RL-based adaptive traffic control for improving transportation efficiency, especially when tailored to the specific conditions of RITI communities.

CHAPTER 1. INTRODUCTION

1.1. Background and Motivation

The advances of artificial intelligence (AI) are transforming the transportation sector, one of the most critical infrastructures in modern society [1]. AI-based technologies have been used in many facets of the transportation system, such as autonomous vehicles, driver injury prediction and prevention, and traffic management. In particular, efficient traffic management can greatly reduce traffic congestion, a problem that we all face on a daily basis. Therefore, it is of paramount importance to develop better traffic management systems, which will in turn boost the efficiency of the overall transportation system.

One effective measure of traffic management is intersection management, where we optimize the phasing of traffic signals at each intersection of the transportation network. Recently, reinforcement learning (RL) has been applied to adaptive traffic signal control and demonstrated superior performance [2–4]. In a nutshell, reinforcement learning is a branch of machine learning and aims at learning to optimally interact with dynamic environments. In the context of adaptive traffic signal control, reinforcement learning algorithms can learn to optimally set the phase of traffic signals under time-varying environments of traffic conditions, given enough training data.

However, the potential of standard reinforcement learning is limited for intersection management in rural, isolated, tribal, and indigenous (RITI) communities due to unique challenges. Specifically, RITI communities are at elevated risks of extreme weather conditions and natural disasters. These extreme events rarely happen, and yet have significant and perhaps disproportionate impact on the performance of the transportation system. Standard reinforcement learning relies heavily on the experiences to learn how to react under such events. But these extreme events do not occur often enough for the reinforcement learning algorithms to learn how to optimally manage the intersections under these rare conditions. Therefore, conventional reinforcement learning algorithms will converge slowly, if converge at all, in the presence of rare extreme events.

1.2. Overview of The Report

In this project, we propose safe reinforcement learning for intelligent traffic signal control for RITI communities under rare extreme events. The key innovation behind safe reinforcement learning under significant rare events is the adjustment of rare event probabilities in the training process. More specifically, we can artificially increase the probabilities of significant rare events (e.g., extreme weather conditions that paralyze the transportation system) in a simulator, such as the Simulation of Urban MObility (SUMO) platform [5]. Then we make proper adjustment (e.g., through importance sampling) in the learning process to account for the altered rare event probabilities [6, 7]. Under well-designed importance sampling techniques, the policy can learn to behave in response to rare events, even though they occur with extremely low probabilities. Based on this simulation principle, we investigate several reinforcement learning paradigms for intersection management, in the quest of improving the robustness and efficiency of the learned policy.

In Chapter 2, we first formulate the intersection management problem as a Markov decision process (MDP), outline our design and analysis framework, and describe the simulation platform.

In Chapter 3, we present our preliminary results on evaluating a variety of deep reinforcement learning algorithms. The purpose of this evaluation process is to select a few high-performing candidates. We build on these candidate algorithms to develop the final proposed algorithm.

In Chapter 4, we describe our main results, namely the design of multi-agent offline RL algorithms for improved efficiency and robustness. We demonstrate the performance improvement of the proposed algorithm through extensive simulations.

In Chapter 5, we discuss some preliminary results in applying federated learning.

We conclude the report in Chapter 6.

CHAPTER 2. DEEP REINFORCEMENT LEARNING FOR INTERSECTION MANAGEMENT

In this chapter, we formulate the problem of optimal intersection management under rare weather events as a Markov decision process (MDP). Then we describe our design and analysis framework that accounts for the rarity of extreme weather events. Finally, we introduce our simulation environments and test cases.

2.1. Problem Formulation as Markov Decision Process

2.1.1. Preliminaries

A Markov decision process can be defined by $\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, r, \gamma, H, \mu_0)$, where $\mathcal{S}$ is the set of states, $\mathcal{A}$ is the set of actions, $P: \mathcal{S} \times \mathcal{A} \rightarrow \Delta(\mathcal{S})$ is the state transition probability function with $\Delta(\mathcal{S})$ being the set of probability distributions over the set $\mathcal{S}$, $r: \mathcal{S} \times \mathcal{A} \rightarrow R$ is the reward function, $\gamma \in [0,1)$ is the discount factor, $H \in N_+$ is the time horizon, and $\mu_0 \in \Delta(\mathcal{S})$ is the initial state distribution. A policy $\pi: \mathcal{S} \rightarrow \Delta(\mathcal{A})$ specifies the action selection probability in state s. Given the MDP $\mathcal{M}$, we would like to find a policy to maximize the expected average discounted reward. This can be formulated as the following optimization problem:

$$\max_\pi (1-\gamma) \cdot E_\pi \left[\sum_{t=0}^{H} \gamma^t r(s_t, a_t) \right].$$

Here, E_π is the expectation with respect to the distribution of trajectories generated by $s_0 \sim \mu_0$, $a_n \sim \pi(\cdot\, | s_t)$, and $s_{n+1} \sim P(\cdot\, | s_n, a_n)$ for all $n = 0, \ldots, H$.

Note that the time horizon can be infinity, namely $H = \infty$.

2.1.2. Problem Formulation

We formulate the intersection management problem as an infinite-horizon Markov decision process. We divide the time into discrete epochs $n = 1,2, \ldots$. At each epoch n, the ingredients of the MDP are defined as follows.

- *State* s_n: The state s_n summarizes all available information relevant to the problem at epoch n. An *example* of the state could include the numbers of vehicles entering each intersection, lanes and driving directions of the vehicles, and current wait time of the vehicles at red light.
- Action a_n: The action a_n is the collection of control variables at epoch n. An *example* of the action could be the phase of each intersection in the network, where the phase is the combination of all signals at each direction of the intersection. For example, the phase “GrGr” at a four-way intersection indicates that the light in four directions are green, red, green, red (e.g., in the counter-clock wise direction).
- *Reward function* $r_n(s_n, a_n)$: The reward function describes the reward we get in epoch n, which depends on the current state and action. A negative reward indicates incurred cost. An *example* reward function could be the negative of the weighted sum of the total wait time of all vehicles and the total queue length at all intersections, where weights can be set to balance different performance criteria (e.g., wait time and queue length).

Note that the above definition serves as an illustrative example. There can be numerous ways of defining states, actions and rewards, depending on the information available, the objective of intersection management. For example, if we have more detailed road conditions and information about individual vehicles, we may include lanes and driving directions of each vehicle in the state. Similarly, we can also define the action as phase switch or phase duration. Last but not the least, we can add additional terms in the reward function to avoid collapse of the system under extreme events.

Since we are concerned with the system under significant rare events, we model the states in more detail. In particular, we can define a subset $\mathcal{T}$ of the entire state space $\mathcal{S}$, where each state $s \in T$ is a rare event. At each state $s \in S$, there is a small probability $\varepsilon(s)$ that the next state will be a rare event state. Therefore, the state transition probability can be written as $p(s'|s,a) = [1 - \varepsilon(s)] \cdot f(s'|s,a)$ if $s' \in \mathcal{T}$ and $p(s'|s,a) = \varepsilon(s) \cdot g(s'|s,a)$ if $s' \notin \mathcal{T}$. Here, $g(s'|s,a)$ and $f(s'|s,a)$ are the state transition probabilities when the next state is a rare event state or not, respectively. We explicitly use different probability transition functions $g(s'|s,a)$ and $f(s'|s,a)$ to model the different system dynamics under regular conditions and under extreme events.

2.2. Safe reinforcement learning under rare events

A reinforcement learning algorithm aims to learn a policy

$$\pi(s,a) = Pr(a_n = a|s_n = s)$$

that specifies the probability distribution of the actions to take under each state. Given a policy π, we can compute the value function, which is the expected reward starting from each state, by solving the following Bellman equation:

$$V^{\pi}(s) = \sum_{a \in A} \pi(s,a) \sum_{s' \in S} p(s'|s,a)[r(s,a) + \gamma V^{\pi}(s')]$$

where $\gamma \in (0,1)$ is a discount factor.

The well-known temporal difference learning algorithm observes the pair (s,a,r,s') at each epoch, and updates the value function as follows:

$$V(s) \leftarrow V(s) + \beta[r + \gamma V(s') - V(s)]$$

where $\beta \in (0,1)$ is the learning rate. It is proved that the temporal difference learning algorithm can converge to the optimal value function, *if each state is visited "sufficiently many times"*. However, since rare events occur with extremely low probabilities, the rare event states may not be visited sufficiently often. Therefore, the fundamental assumption to make the TD learning work does not hold when there are rare events. The learned policy would "ignore" the rare events and fail to learn how to behave in these situations. We can expect the same "curse of rarity" in other RL algorithms. Fig. 2.1 illustrates this challenge.

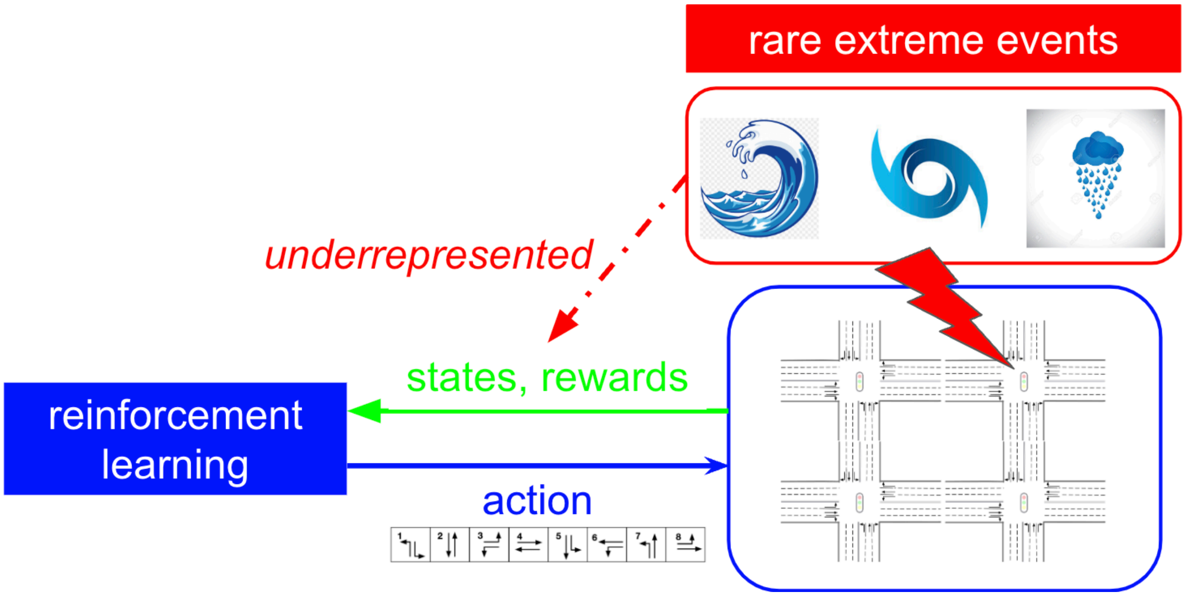


Figure 2.1 Challenges of reinforcement learning for intersection management under rare events. The low-probability extreme events are underrepresented in the data samples of states and rewards. Therefore, it is difficult for the algorithm to learn how to behave under these states.

2.3. Importance Sampling Techniques

One way to deal with the challenges of rare events is to artificially increase the probabilities of rare events in simulation and use importance sampling techniques [6, 7] to correctly learn the true value function.

Mathematically, this involves increasing the probabilities of the significant rare events to $\hat{\varepsilon}(s) > \varepsilon(s)$ during the training phase, and adjusting the value function update equation (e.g., temporal difference learning) using importance sampling. The framework is illustrated in Fig. 2.2.

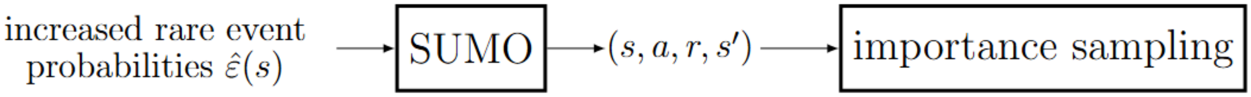


Figure 2.2 Illustration of the importance sampling technique to deal with the rarity of extreme events.

2.4. Simulation Platform and Experiment Setup

In this project, we use the widely-used SUMO platform for simulating a network of intersections. See Fig. 2.3 for an example of a four-way intersection in SUMO.

We build on two open-source repositories, SUMO-RL [8] and Reinforcement Learning Benchmarks for Traffic Signal Control (RESCO) [9], which provide the interface between reinforcement learning repositories and the SUMO simulator.

Throughout the simulation, we use the following specification of the MDP.

- States:
    - the current active green phase;
    - a boolean variable of whether a given amount of time (in seconds) have already passed in the current phase;
    - lane density: the number of vehicles in incoming lane divided by the total capacity of the lane;
    - lane queues: the number of queued (speed below 0.1 m/s) vehicles in incoming lane divided by the total capacity of the lane.
- Actions: the next phase of the intersection.
- Rewards: the negative of the total delay (i.e., the total delay is minimized when the reward is maximized).

The probability of the rare event $\varepsilon(s)$ is a parameter that will be varied for different evaluation scenarios. The state transition probability depends on the traffic conditions (e.g., distribution of incoming vehicles). We use the default settings in the SUMO platform.

2.5. State of The Art

There has been extensive research on intersection management [10]. Researchers have utilized and developed different techniques, such as control theory [11, 12], optimization [13]), heuristics [14], and hybrid of the above techniques [10].

Reinforcement learning has found its success in a variety of areas, such as learning to play Atari games [15] and Go [16]. It has also been applied to various research areas in transportation, such as control of autonomous vehicle [17, 18], fleet management [19–21], and routing [22]. Recently, there are works that apply reinforcement learning to intersection management [2–4]. These works use standard reinforcement learning to adaptively control traffic signals in order to optimize the performance (in terms of delay, throughput, etc.) of the transportation system. However, these works did not consider significant rare events in the transportation system and their impact on the performance of reinforcement learning.

Reinforcement learning under rare events were proposed in the generic settings [6, 7]. But these general frameworks may not consider specific features of the transportation system and can be improved when applying to intersection management.

To the best of our knowledge, there has been no work that applies safe reinforcement learning under significant rare events to intersection management. The development of such algorithms is crucial for the RITI communities.

In the next two chapters, we will first present our preliminary results on evaluating existing RL algorithms on our problem setting, and then describe the proposed algorithm with improved efficiency and robustness.

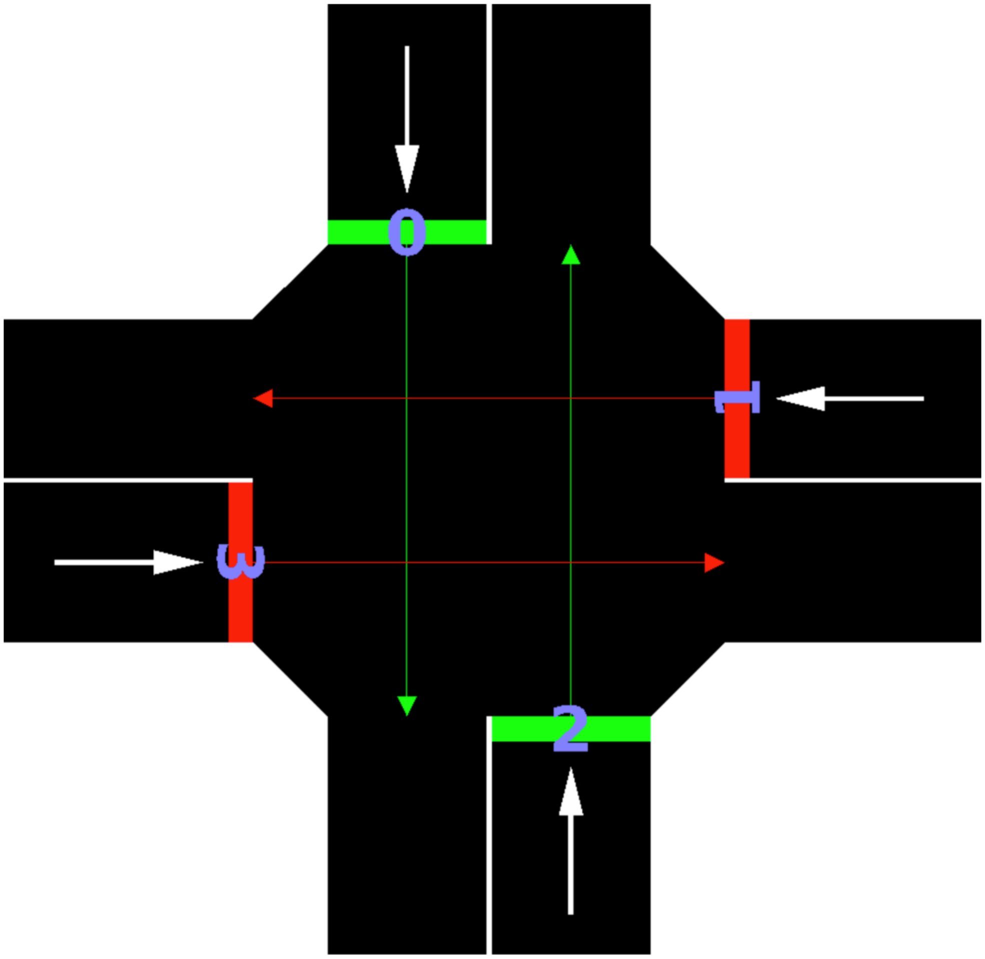


Figure 2.3 Illustration of a four-way intersection in the SUMO platform [5]. The current phase of this intersection is GrGr.

CHAPTER 3. PRELIMINARY RESULTS

In this chapter, we present preliminary results on the evaluation of existing RL algorithms under two synthetic evaluation cases. The road networks in these two cases are much smaller than the test cases in Chapter 4. However, they provide important insights and help us narrow down our search and design of the final algorithm. More concretely, this evaluation process is necessary due to several reasons.

- The variety of available RL algorithms: Due to convenient interfaces provided by SUMO-RL [8] and RESCO [9], we can train and evaluate all the RL algorithms available on stable-baselines3 [23] and RLlib [24]. While this offers great flexibility, it would also incur a high time cost if we were to train and test all available algorithms in the test cases. Hence, it is important to weed out the algorithms that are unlikely to perform well in the test cases.
- Complexity of the test cases: As we will discuss in Chapter 4, the test cases are realistic traffic networks in Cologne and Ingolstadt in the SUMO platform. These test cases have up to 19 intersections, resulting in large state spaces and action spaces. Therefore, it is helpful to focus on algorithms that are more likely to perform well.
- Hyperparameter tuning: The RL algorithms are well known to be sensitive to the hyperparameters (e.g., the learning rate, the neural network architecture of the actor and critic networks) [25]. To ensure that all the RL algorithms are performing, we need to search through the space of hyperparameters. It is impractical to perform hyperparameter tuning for all the available algorithms on all the test cases. As a result, it is useful to limit the number of candidate algorithms for the test cases.

3.1. Evaluation Cases

We use two 4×4 grid networks as the evaluation cases during this initial screening phase. See Fig. 3.1 for illustration of these two evaluation cases.

- “4x4 Grid - 3 Lanes” [26]: A 4x4 grid network with 16 four-way intersections. Each direction has 3 lanes going left, straight, and right, respectively.
- “4x4 Grid - 2 Lanes” [27]: A 4x4 grid network with 16 four-way intersections. Each direction has at most 2 lanes: the horizontal directions have 2 lanes, and the vertical directions have 1 lane.

The traffic data are synthetic data generated according to some distributions. Please see [26, 27] for details.

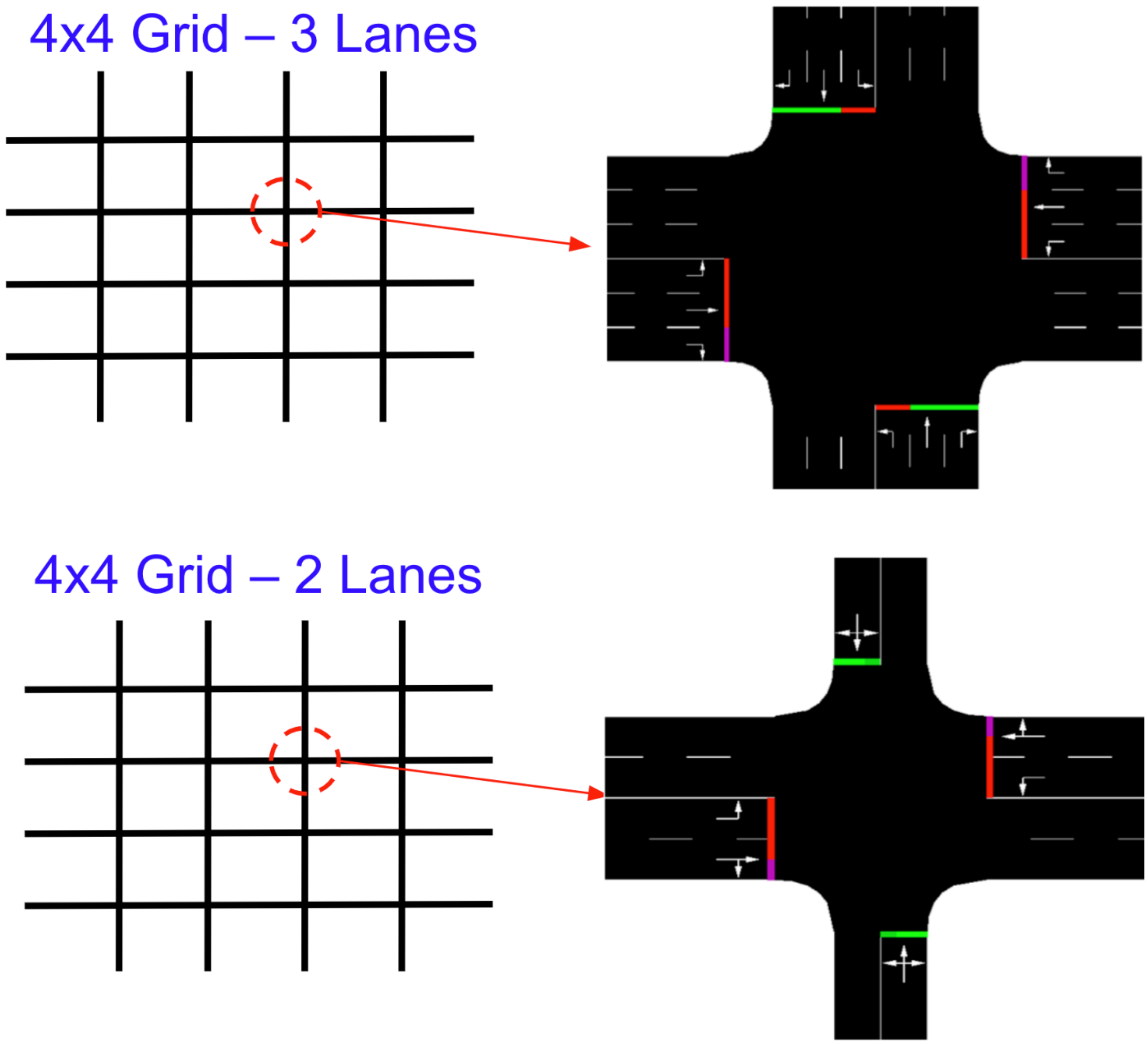


Figure 3.1 Two 4x4 grid networks for initial screening of RL algorithms. Both networks have 16 four-way intersections. One network has 3 lanes in each direction, and the other has 2 lanes in horizontal directions and 1 lane in vertical directions.

3.2. Evaluation Results

We evaluate the RL algorithms in stable-baselines3 that support discrete actions. Below is a brief description of these algorithms.

- **Deep Q-Network (DQN)** [15] - DQN combines the classic Q-learning with deep neural networks, allowing it to handle high-dimensional input spaces such as images. It approximates the Q-value function using a neural network, and employs techniques such as experience replay (sampling previous experiences) and target networks (a more stable Q-value target) to stabilize learning. DQN is one of the first deep reinforcement learning algorithms. It was originally designed to work on problems with discrete actions.

- **Quantile Regression DQN (QR-DQN)** [28] - As an extension of DQN, QR-DQN estimates the distribution of future rewards, as opposed to estimating the mean of the rewards in DQN. By predicting multiple quantiles of the reward distribution, it allows for a more robust performance by taking into account the uncertainty. This is especially beneficial for highly stochastic environments.
- **Advantage Actor-Critic (A2C)** [29] - A2C is an actor-critic algorithm, which uses an actor (policy) and a critic (value function) to estimate both the policy and the value of states. A2C is the synchronous version of the actor-critic algorithms where multiple agents (actors) are trained in parallel. It uses an advantage function (which measures how good an action is compared to others) to update the actor, stabilizing learning by reducing variance in policy updates.
- **Augmented Random Search (ARS)** [30] - ARS is a gradient-free method that performs random search in the parameter space of the policies. It evaluates the performance of several random policy variations, selects the best-performing search directions, and updates the main policy based on these variations. Its main advantage is computational efficiency, since it does not require backpropagation to calculate the gradient. However, its performance might not be the best if we allow all the algorithms to run for sufficiently long time.
- **Trust Region Policy Optimization (TRPO)** [31] - TRPO is a policy gradient method that ensures stable updates by enforcing a constraint on the step size of policy updates. This constraint ensures that the new policy doesn't diverge too much from the old one by limiting the change in the Kullback-Leibler (KL) divergence between the two. TRPO is designed to guarantee monotonic improvement in policy performance and is effective in high-dimensional continuous control tasks.
- **Proximal Policy Optimization (PPO)** [32] - PPO is another policy gradient method that improves stability and sample efficiency (empirically) of TRPO by using a novel clipped objective function. It operates in the actor-critic style, and performs stochastic gradient ascent on a "surrogate" objective function that reflects the performance of the policy over multiple epochs.

Table 3.1 summarizes the performance of all the RL algorithms listed above. For each algorithm, we perform hyperparameter optimization, where we randomly choose 50 sets of hyperparameters (i.e., learning rate, coefficient of policy entropy, neural network architecture, etc.). The best results, measured by average delay over 10 random experiments, are reported for each algorithm. We also include the standard deviation for each algorithm. Note that the delay values under the 4x 4 Grid - 2 Lanes scenario is much higher compared to the 3-lane scenario, because of the reduced number of lanes.

Below are some observations from this evaluation stage.

- Out of the six algorithms tested, DQN and PPO are the best. Each one of them achieves the lowest average delay in one evaluation case.
- QR-DQN is not as good as DQN in the evaluation cases. It does have slightly smaller standard deviation as promised, but higher averages. The reason may be that it is less sample efficient.
- As expected, TRPO achieves higher delay than its improved version in PPO. Therefore, we will use PPO instead of TRPO in the test cases.

- ARS and A2C are consistently worse than the other algorithms, showing that random search (ARS) and vanilla on-policy algorithms (A2C) may not work well.

We include details of hyperparameter optimization in Appendix A.

Now that we have identified DQN and PPO as the two best algorithms, we will move on to algorithm design on the test cases based on these two algorithms.

Table 3.1 Performance of RL algorithms on evaluation cases. The metric is delay (in seconds). For each data point, we run 10 random trials and report the average value and the standard deviation. For each algorithm, we show the best results obtained from the one with the best hyperparameters.

	4x4 Grid – **3** Lanes	4x4 Grid – **2** Lanes
ARC	50.0 ± 1.3	135.6 ± 3.2
A2C	47.3 ± 0.8	117.8 ± 2.3
DQN	**21.2** ± **0.6**	83.7 ± 2.9
QR-DQN	33.0 ± 0.5	100.7 ± 1.4
PPO	22.1 ± 0.6	**80.8** ± **2.2**
TRPO	26.3 ± 0.5	95.6 ± 2.0

CHAPTER 4. MULTI-AGENT OFFLINE RL

4.1. Motivation

In Chapter 3, we have demonstrated that (standard) RL has the potential to perform well for intersection management. However, standard RL relies on the *online interactions* with the traffic network for the agents to learn the optimal decision-making rules. More specifically, a RL agent starts with a suboptimal, or even random, policy, and interacts with the traffic network by taking actions, receiving rewards, and observing the next state. Through such interactions, the agent learns how “good” the actions and the policies have been, and keeps refining the policy until it converges to the optimal one. The major drawback is that online interactions are expensive, and will cause significant delay before the algorithm learns the optimal policy.

To tackle this challenge, we propose to use an emerging paradigm of RL, namely **offline RL** [33-37]. Offline RL is fundamentally different from traditional RL, which requires extensive online interactions with the traffic network. In offline RL, **the agent trains on the existing data exclusively without interacting with the environment and could still perform well when deployed**. The terms “offline” emphasize the fact that the RL agent relies on the existing data only and that the training is offline (see Fig. 4.1 for illustration of the distinctions from standard RL). Therefore, we can utilize the already collected data of traffic networks, even though the data may be generated from suboptimal intersection management policies.

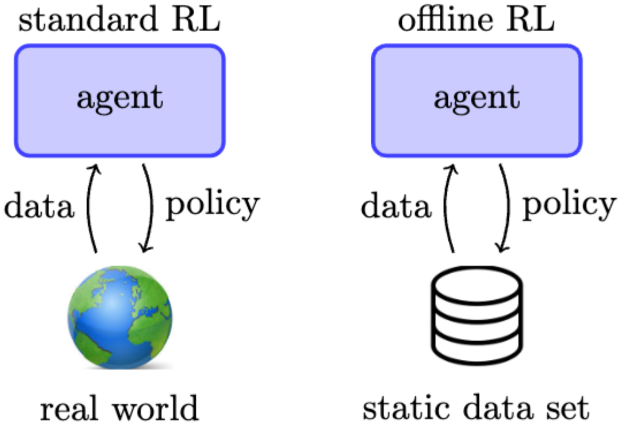


Figure 4.1 Differences between standard reinforcement learning (Chapter 3) and offline RL. In standard RL, the agent *interacts with* the environment by rolling out the policy and *collecting data* from the environment. In offline RL, the agent trains the policy with *existing data* and *does not interact* with the environment.

The goal of this chapter is to develop offline RL algorithms that can train on existing datasets before real-world deployment and generalize well to the real-world environment. In addition, we implement multi-agent versions of offline RL algorithms to further improve the sample efficiency and reduce the computational complexity.

4.2. Limitation of Existing Works on Offline RL

4.2.1. Off-Policy RL on Static Datasets

Offline RL could be viewed as off-policy RL [38–47] on static datasets. Off-policy RL refers to a RL training paradigm in which the agent learns from off-policy data (i.e., data generated from previous iterations of policies). Standard off-policy RL still requires online interactions with the environment, and maintains an adaptive dataset that contains entries of past interactions with the environment. Past works have shown that when directly applied to static datasets, off-policy RL suffers from large bootstrapping errors due to distribution shift (i.e., the values of state-action p airs where is little or no data may be highly inaccurate) [38, 39]. Therefore, standard off-policy RL performs poorly on static datasets [38, 39].

4.2.2. Off-Policy RL on Static Datasets

Behavior cloning. Most existing offline RL works focus on reducing bootstrapping errors due to distribution shift. Since distribution shift arises from the discrepancy between the data-generating policies and the current policies in the training, a natural way to reduce distribution shift is imitation learning or behavioral cloning, namely to mimic the policies that generated the dataset [48–51]. A fundamental limitation of behavioral cloning is that the data-generating policy may be highly suboptimal, setting a low ceiling for the offline policies.

Model-based approach. Some offline RL works adopt the model-based RL approach [52–54]. In model-based RL, the agent first learns the underlying Markov decision process (MDP) by estimating state transition probabilities and rewards from the data. Given the learned model, it uses dynamic programming to solve for the optimal policy. In the offline setting, it is typical to learn a pessimistic MDP, where the reward is penalized by the uncertainty from the dataset (e.g., adding a penalty for the state-action pairs uncommon or unseen in the dataset) [55–68]. However, the prevalent pessimistic approach in model-based offline RL can lead to overly conservative models, resulting in under-performing policies.

Model-free approach. Some offline RL works adopt the model-free RL approach, where the agent directly learns the Q-values of the state-action pairs (the expected cumulative rewards starting from the given state-action pair) and the optimal policy from the Q values. To deal with distribution shift in the offline setting, existing works [38, 39, 69–95] have proposed to (1) restrict the action space so that the learned policy is close to the data-generating policy (e.g., imposing some divergence constraints [38, 80, 81]), (2) learn a pessimistic Q-value function (e.g., augmenting the standard Bellman error objective with a Q-value regularizer [74, 78]), or (3) use data augmentation [92]. Similar to model-based offline RL, pessimism is shown to be important for model-free offline RL [83, 93], leading to conservative policies.

Theoretical Analysis of Sample Complexity. There are theoretical works on how many samples are needed in the static dataset for offline RL to learn the optimal policy in the online setting [41, 96–108]. It is proven that the sample efficiency is exponential in the time horizon of the MDP under mild conditions [105], and that to achieve polynomial sample efficiency, we need strong assumptions on the data coverage [108]. For example, the state-of-the-art result requires the data-generating policy to have taken the optimal action at each state (possibly with incorrect probabilities) and to have visited all the possible states [108]. While such requirements guarantee polynomial sample complexity to achieve the optimal policy, they may be unrealistic in practice.

4.3. Test Cases

4.3.1. Test Scenarios

We use realistic traffic networks in the cities of Cologne and Ingolstadt as the test cases. These two cases are widely used SUMO scenarios "TAPAS Cologne" [109] and "InTAS" [110]. See Fig. 4.2 for illustration of these two test cases (**Figure credit: RESCO Github Repository** [9]).

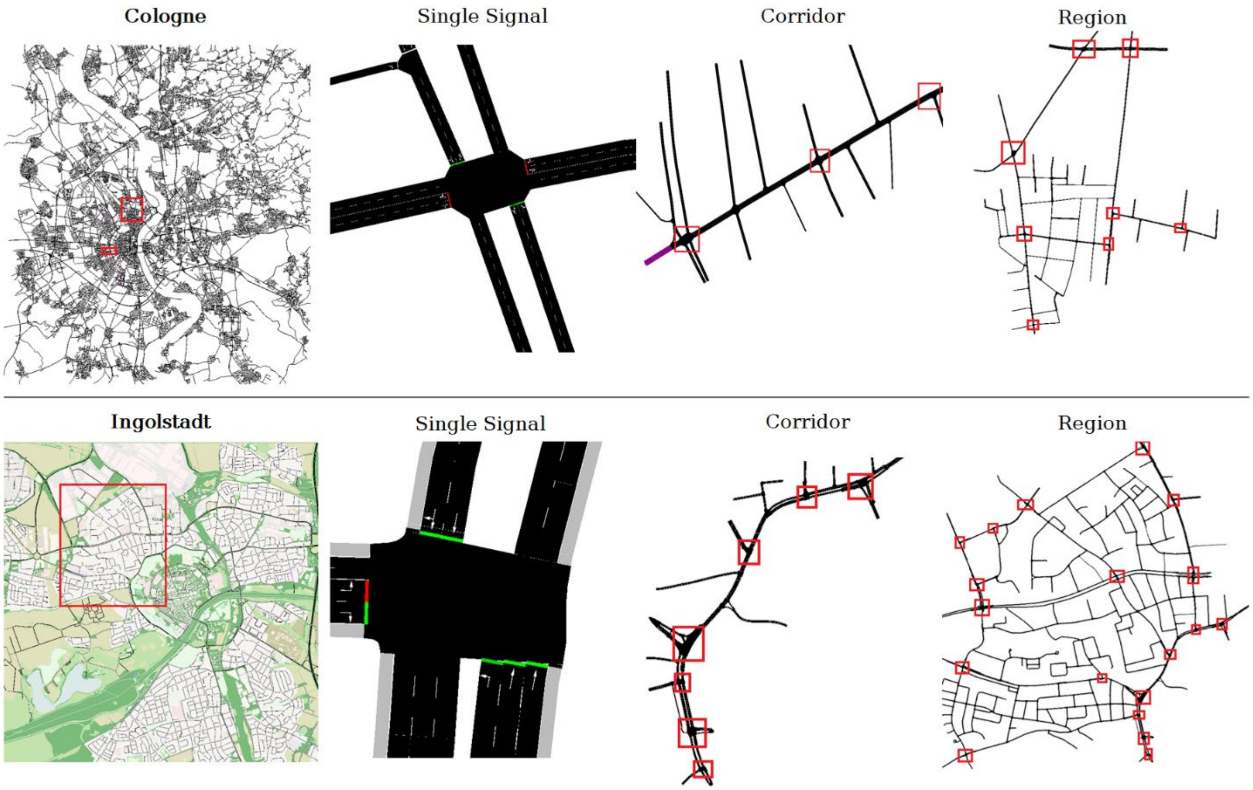


Figure 4.2 Illustration of the Cologne and Ingolstadt networks as the test cases. In each network, we have three test scenarios: a single intersection ("Single Signal"), a road with multiple intersections ("Corridor"), and a network with multiple roads and intersections ("Region"). Figure credit: RESCO Github Repository [9].

In each network, there are three test scenarios.

- "Single Signal": A single signalized intersection.
- "Corridor": A main road with multiple signalized intersections along the road.
- "Region": A network of multiple roads and multiple signalized intersections.

The traffic data is provided by SUMO, which is derived from measurement and estimation of real traffic data in these two cities. In summary, there are six test scenarios, where each test case has three scenarios of "Single Signal", "Corridor", and "Region".

4.3.2. Offline Datasets

We use realistic traffic networks in the cities of Cologne and Ingolstadt as the test cases. These two cases are widely used SUMO scenarios “TAPAS Cologne” [109] and “InTAS” [110]. See Fig. 4.2 for illustration of these two test cases (**Figure credit: RESCO Github Repository** [9]).

In offline RL, the offline dataset, which the offline RL algorithm trains on, has a significant impact on the performance. We generate five datasets for each of the six test scenarios.

- **Fixed-Time**: a dataset collected by running the default fixed phase duration in SUMO;
- **Max-Pressure**: a dataset generated by running the max joint pressure algorithm [26];
- **Greedy**: a dataset generated by running the max joint queue length and vehicle count algorithm [27];
- **Expert-DQN**: a dataset generated by running the DQN algorithm in Chapter 3;
- **Expert-PPO**: a dataset generated by running the PPO algorithm in Chapter 3.

Note that we call the last two datasets “expert” because they are generated by optimized RL algorithms.

4.3.3. Offline RL Algorithms

We test two offline RL algorithms, Behavior Cloning (**BC**) and Monotonic Advantage Re-Weighted Imitation Learning (**MARWIL**) [48]. Below is a brief description of these two algorithms

- **Behavior Cloning (BC)** - Behavior cloning learns the policy directly from a dataset of expert trajectories by mimicking the expert’s actions. In other words, it treats the RL problem as a supervised learning task, where the data is the state and the label is the action. BC aims to predict the action given the state with high accuracy. No reward is needed in BC.
- **Monotonic Advantage Re-Weighted Imitation Learning** (**MARWIL**) [48] - MARWIL is an extension of behavior cloning by incorporating the reward signal. More specifically, it weights the action by an advantage function that reflects the reward of this state-action pair. As a result, it puts higher weights on the trajectories that yield higher rewards.

We chose these two algorithms because they are classic offline RL algorithms with stable implementations in Rllib [24].

4.4. Test Results

Table 4.1 summarizes the performance of the offline RL algorithms under all six test scenarios and all five offline datasets. We evaluate each algorithm under 30 combinations of test scenarios and offline datasets. All delay values are the average over 10 random trials.

Below are key observations from our extensive performance evaluation.

- Overall, we can observe an increased delay compared to online RL algorithms reported in [9]. This is expected because offline RL algorithms do not interact with the network online.
- The quality of the offline datasets has a large impact on the performance of the offline RL algorithms. Both offline RL algorithms perform much better when trained on datasets generated from expert policies (DQN and PPO). This is reasonable, because the two offline RL algorithms

are basically imitation learning algorithms and aim to mimic the behavior of the policy that generated the dataset.

- MARWIL performs slightly better than BC in the majority of the scenarios (24 out of 30 scenarios). However, the differences are smaller (usually around 1-2 seconds). Given that the values are averaged over 10 trials only, the statistical significance may not be high enough to make a definitive conclusion that MARWIL is better than BC.

Table 4.1 Performance of offline RL algorithms on test cases. The metric is delay (in seconds). For each data point, we run 10 random trials and report the average value. For each algorithm, we show the best results obtained from the one with the best hyperparameters.

<table>
<tr><th colspan="6">Cologne – Single Signal</th></tr>
<tr><th></th><th>Fixed-Time</th><th>Max-Pressure</th><th>Greedy</th><th>Expert-DQN</th><th>Expert-PPO</th></tr>
<tr><td>BC</td><td>68.1</td><td>51.4</td><td><b>74.7</b></td><td>39.3</td><td>69.4</td></tr>
<tr><td>MARWIL</td><td><b>67.3</b></td><td><b>49.2</b></td><td>75.3</td><td><b>36.9</b></td><td><b>67.4</b></td></tr>
<tr><th colspan="6">Cologne – Corridor</th></tr>
<tr><th></th><th>Fixed-Time</th><th>Max-Pressure</th><th>Greedy</th><th>Expert-DQN</th><th>Expert-PPO</th></tr>
<tr><td>BC</td><td>60.3</td><td>107.3</td><td>182.5</td><td>31.2</td><td>33.4</td></tr>
<tr><td>MARWIL</td><td><b>60.1</b></td><td><b>102.5</b></td><td><b>177.2</b></td><td><b>30.1</b></td><td><b>31.7</b></td></tr>
<tr><th colspan="6">Cologne – Region</th></tr>
<tr><th></th><th>Fixed-Time</th><th>Max-Pressure</th><th>Greedy</th><th>Expert-DQN</th><th>Expert-PPO</th></tr>
<tr><td>BC</td><td>81.8</td><td><b>38.2</b></td><td>64.1</td><td>30.6</td><td>31.9</td></tr>
<tr><td>MARWIL</td><td><b>79.1</b></td><td>40.5</td><td><b>62.9</b></td><td><b>30.3</b></td><td><b>30.2</b></td></tr>
<tr><th colspan="6">Ingolstadt – Single Signal</th></tr>
<tr><th></th><th>Fixed-Time</th><th>Max-Pressure</th><th>Greedy</th><th>Expert-DQN</th><th>Expert-PPO</th></tr>
<tr><td>BC</td><td>56.3</td><td>48.6</td><td><b>30.2</b></td><td>30.9</td><td>28.8</td></tr>
<tr><td>MARWIL</td><td><b>55.3</b></td><td><b>47.3</b></td><td>30.9</td><td><b>29.1</b></td><td><b>27.9</b></td></tr>
<tr><th colspan="6">Ingolstadt – Corridor</th></tr>
<tr><th></th><th>Fixed-Time</th><th>Max-Pressure</th><th>Greedy</th><th>Expert-DQN</th><th>Expert-PPO</th></tr>
<tr><td>BC</td><td>128.7</td><td>101.6</td><td>70.3</td><td>44.8</td><td><b>42.1</b></td></tr>
<tr><td>MARWIL</td><td><b>125.8</b></td><td><b>99.6</b></td><td><b>69.3</b></td><td><b>42.9</b></td><td>43.1</td></tr>
<tr><th colspan="6">Ingolstadt – Region</th></tr>
<tr><th></th><th>Fixed-Time</th><th>Max-Pressure</th><th>Greedy</th><th>Expert-DQN</th><th>Expert-PPO</th></tr>
<tr><td>BC</td><td>182.6</td><td>179.4</td><td>99.5</td><td><b>78.4</b></td><td><b>83.2</b></td></tr>
<tr><td>MARWIL</td><td><b>180.1</b></td><td><b>177.6</b></td><td><b>98.3</b></td><td>79.8</td><td>84.3</td></tr>
</table>

CHAPTER 5. PRELIMINARY RESULTS ON FEDERATED LEARNING

Federated learning (FL) is an emerging technique that enables multiple agents to collaboratively train a machine learning model **with local data only**. In other words, the agents do not share data, resulting in a major advantage of FL in preserving data privacy. In many RITI communities, data privacy and ownership are critical concerns. Centralizing traffic data for RL training might face resistance due to legal, cultural, or privacy reasons. By using federated learning, data can remain on local servers at each community's intersection, while only model updates are shared across locations. This approach mitigates privacy risks while still allowing for collaboration to improve the global model.

In our preliminary results [111], we proposed a novel distributed stochastic gradient descent method for federated learning. Our method sparsifies each gradient descent step to optimize the convergence performance by balancing the trade-off between communication cost and convergence error. It performed well on general image classification tasks using the MNIST, CIFAR-10 datasets. We also demonstrated that the proposed adaptive Top-K algorithm in SGD achieves a significantly better convergence rate compared to state-of-the-art methods on multi-robot collaboration tasks.

It would be an interesting future direction to study federated reinforcement learning in intersection management.

CHAPTER 6. CONCLUSION

This study investigated the use of reinforcement learning for adaptive traffic signal control, with a focus on addressing the unique challenges faced by RITI communities, such as extreme weather events. Our evaluations revealed that DQN and PPO outperformed other algorithms in minimizing traffic delays, while offline RL algorithms like MARWIL and BC showed improved performance when trained on high-quality datasets from expert policies. The results underscore the importance of using advanced RL algorithms and high-quality training data to address the complexities of traffic management in challenging environments.

We would like to note several limitations of our study and directions of future works.

- It is well known that the performance of RL algorithms are highly stochastic [112]. In our experiments, we use a limited number (10) of random trials due to the large number of test scenarios. In the future study, it is important to perform more comprehensive experiments to ensure the statistical significance of our findings. One promising approach is to integrate our code with existing repositories (e.g., rliable [112]) that specializes in performance analysis of RL algorithms.
- We tested two offline RL algorithms, both of which are based on imitation learning. It is crucial to test other recent offline RL algorithms that aim to surpass, instead of mimicking, the policy that generates the offline dataset. We can integrate with emerging repositories with implementation of recent offline RL algorithms, such as d3rlpy [113] and CORL [114].
- It is important to establish a set of well-recognized datasets and performance benchmarks to study the performance of offline RL algorithms in intersection management.

REFERENCES

[1] Yuxi Li. Deep reinforcement learning: An overview. arXiv preprint arXiv:1701.07274, 2017.
[2] Samah El-Tantawy, Baher Abdulhai, and Hossam Abdelgawad. Multiagent reinforcement learning for integrated network of adaptive traffic signal controllers (MARLINATSC): methodology and large-scale application on downtown toronto. IEEE Transactions on Intelligent Transportation Systems, 14(3):1140–1150, 2013.
[3] Tianshu Chu, Jie Wang, Lara Codec`a, and Zhaojian Li. Multi-agent deep reinforcement learning for large-scale traffic signal control. IEEE Transactions on Intelligent Transportation Systems, 2019.
[4] Hua Wei, Guanjie Zheng, Huaxiu Yao, and Zhenhui Li. Intellilight: A reinforcement learning approach for intelligent traffic light control. In Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pages 2496–2505. ACM, 2018.
[5] Pablo Alvarez Lopez, Michael Behrisch, Laura Bieker-Walz, Jakob Erdmann, Yun-Pang Flotterod, Robert Hilbrich, Leonhard Lucken, Johannes Rummel, Peter Wagner, and Evamarie Wiesner. Microscopic traffic simulation using sumo. In The 21st IEEE International Conference on Intelligent Transportation Systems. IEEE, 2018.
[6] Jordan Frank, Shie Mannor, and Doina Precup. Reinforcement learning in the presence of rare events. In Proceedings of the 25th international conference on Machine learning, pages 336–343, 2008.
[7] Kamil Andrzej Ciosek and Shimon Whiteson. Offer: Off-environment reinforcement learning. In Thirty-first AAAI conference on artificial intelligence, 2017.
[8] Lucas N. Alegre. SUMO-RL. https://github.com/LucasAlegre/sumo-rl, 2019.
[9] James Ault and Guni Sharon. Reinforcement learning benchmarks for traffic signal control. In Proceedings of the Thirty-fifth Conference on Neural Information Processing Systems (NeurIPS 2021) Datasets and Benchmarks Track, December 2021.
[10] Elnaz Namazi, Jingyue Li, and Chaoru Lu. Intelligent intersection management systems considering autonomous vehicles: a systematic literature review. IEEE Access, 7:91946–91965, 2019.
[11] Andreas A Malikopoulos, Christos G Cassandras, and Yue J Zhang. A decentralized energy-optimal control framework for connected automated vehicles at signal-free intersections. Automatica, 93:244–256, 2018.
[12] Huifu Jiang, Jia Hu, Shi An, Meng Wang, and Byungkyu Brian Park. Eco approaching at an isolated signalized intersection under partially connected and automated vehicles environment. Transportation Research Part C: Emerging Technologies, 79:290–307, 2017.
[13] Pengfei Taylor Li and Xuesong Zhou. Recasting and optimizing intersection automation as a connected-and-automated-vehicle (CAV) scheduling problem: A sequential branch-and-bound search approach in phase-time-traffic hypernetwork. Transportation Research Part B: Methodological, 105:479–506, 2017.
[14] Guni Sharon and Peter Stone. A protocol for mixed autonomous and human-operated vehicles at intersections. In International Conference on Autonomous Agents and Multiagent Systems, pages 151–167. Springer, 2017.
[15] Volodymyr Mnih. Playing Atari with deep reinforcement learning. arXiv preprint arXiv:1312.5602, 2013.
[16] David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, et al. A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play. Science, 362(6419):1140–1144, 2018.

[17] Xuewei Qi, Yadan Luo, GuoyuanWu, Kanok Boriboonsomsin, and Matthew J Barth. Deep reinforcement learning-based vehicle energy efficiency autonomous learning system. In 2017 IEEE Intelligent Vehicles Symposium (IV), pages 1228–1233. IEEE, 2017.
[18] Yukimasa Matsumoto and Kazuya Nishio. Reinforcement learning of driver receiving traffic signal information for passing through signalized intersection at arterial road. Transportation Research Procedia, 37:449–456, 2019.
[19] Hugo P Simao, Jeff Day, Abraham P George, Ted Gifford, John Nienow, and Warren B Powell. An approximate dynamic programming algorithm for large-scale fleet management: A case application. Transportation Science, 43(2):178–197, 2009.
[20] Kaixiang Lin, Renyu Zhao, Zhe Xu, and Jiayu Zhou. Efficient large-scale fleet management via multi-agent deep reinforcement learning. In Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pages 1774–1783. ACM, 2018.
[21] Suining He and Kang G Shin. Spatio-temporal capsule-based reinforcement learning for mobility-on-demand network coordination. In Proceedings of the 2019 Web Conference. ACM, 2019.
[22] Abdul Rahman Kreidieh, Cathy Wu, and Alexandre M Bayen. Dissipating stop and go waves in closed and open networks via deep reinforcement learning. In 2018 21st International Conference on Intelligent Transportation Systems (ITSC), pages 1475–1480. IEEE, 2018.
[23] Antonin Raffin, Ashley Hill, Adam Gleave, Anssi Kanervisto, Maximilian Ernestus, and Noah Dormann. Stable-Baselines3: Reliable Reinforcement Learning Implementations. Journal of Machine Learning Research, 22(268):1–8, 2021.
[24] Eric Liang, Richard Liaw, Robert Nishihara, Philipp Moritz, Roy Fox, Ken Goldberg, Joseph Gonzalez, Michael Jordan, and Ion Stoica. RLlib: Abstractions for distributed reinforcement learning. In Jennifer Dy and Andreas Krause, editors, Proceedings of the 35th International Conference on Machine Learning, volume 80 of Proceedings of Machine Learning Research, pages 3053–3062. PMLR, 10–15 Jul 2018.
[25] Logan Engstrom, Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Firdaus Janoos, Larry Rudolph, and Aleksander Madry. Implementation matters in deep RL: A case study on PPO and TRPO. In International Conference on Learning Representations, 2020.
[26] 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.
[27] Jinming Ma and Feng Wu. Feudal multi-agent deep reinforcement learning for traffic signal control. In Proceedings of the 19th international conference on autonomous agents and multiagent systems (AAMAS), pages 816–824, 2020.
[28] Will Dabney, Mark Rowland, Marc Bellemare, and R´emi Munos. Distributional reinforcement learning with quantile regression. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 32, 2018.
[29] Volodymyr Mnih. Asynchronous methods for deep reinforcement learning. arXiv preprint arXiv:1602.01783, 2016.
[30] Horia Mania, Aurelia Guy, and Benjamin Recht. Simple random search provides a competitive approach to reinforcement learning. arXiv preprint arXiv:1803.07055, 2018.
[31] John Schulman, Sergey Levine, Philipp Moritz, Michael Jordan, and Pieter Abbeel. Trust region policy optimization. arXiv preprint arXiv:1502.05477, 2015.

[32] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. arXiv preprint arXiv:1707.06347, 2017.
[33] Sergey Levine, Aviral Kumar, George Tucker, and Justin Fu. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. arXiv preprint arXiv:2005.01643, 2020.
[34] Louis Monier, Jakub Kmec, Alexandre Laterre, Thomas Pierrot, Valentin Courgeau, Olivier Sigaud, and Karim Beguir. Offline reinforcement learning hands-on. arXiv preprint arXiv:2011.14379, 2020.
[35] Justin Fu, Aviral Kumar, Ofir Nachum, George Tucker, and Sergey Levine. D4RL: Datasets for deep data-driven reinforcement learning. arXiv preprint arXiv:2004.07219, 2020.
[36] Caglar Gulcehre, Ziyu Wang, Alexander Novikov, Thomas Paine, Sergio Gomez, Konrad Zolna, Rishabh Agarwal, Josh S Merel, Daniel J Mankowitz, Cosmin Paduraru, Gabriel Dulac-Arnold, Jerry Li, Mohammad Norouzi, Matthew Hoffman, Nicolas Heess, and Nando de Freitas. RL unplugged: A suite of benchmarks for offline reinforcement learning. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, Advances in Neural Information Processing Systems, volume 33, pages 7248–7259. Curran Associates, Inc., 2020.
[37] Rongjun Qin, Songyi Gao, Xingyuan Zhang, Zhen Xu, Shengkai Huang, Zewen Li, Weinan Zhang, and Yang Yu. NeoRL: A near real-world benchmark for offline reinforcement learning. arXiv preprint arXiv:2102.00714, 2021.
[38] Aviral Kumar, Justin Fu, Matthew Soh, George Tucker, and Sergey Levine. Stabilizing off-policy Q-learning via bootstrapping error reduction. Advances in Neural Information Processing Systems, 32:11784–11794, 2019.
[39] Scott Fujimoto, David Meger, and Doina Precup. Off-policy deep reinforcement learning without exploration. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, Proceedings of the 36th International Conference on Machine Learning, volume 97 of Proceedings of Machine Learning Research, pages 2052–2062. PMLR, 09–15 Jun 2019.
[40] Nathan Kallus and Masatoshi Uehara. Intrinsically efficient, stable, and bounded off-policy evaluation for reinforcement learning. Advances in Neural Information Processing Systems, 32:3325–3334, 2019.
[41] Masatoshi Uehara, Masaaki Imaizumi, Nan Jiang, Nathan Kallus, Wen Sun, and Tengyang Xie. Finite sample analysis of minimax offline reinforcement learning: Completeness, fast rates and first-order efficiency. arXiv preprint arXiv:2102.02981, 2021.
[42] Nathan Kallus and Masatoshi Uehara. Doubly robust off-policy value and gradient estimation for deterministic policies. Advances in Neural Information Processing Systems, 33, 2020.
[43] Masatoshi Uehara, Masahiro Kato, and Shota Yasui. Off-policy evaluation and learning for external validity under a covariate shift. In NeurIPS, 2020.
[44] Tengyu Xu, Zhuoran Yang, Zhaoran Wang, and Yingbin Liang. Doubly robust offpolicy actor-critic: Convergence and optimality. arXiv preprint arXiv:2102.11866, 2021.
[45] Nathan Kallus and Masatoshi Uehara. Statistically efficient off-policy policy gradients. In International Conference on Machine Learning, pages 5089–5100. PMLR, 2020.
[46] Masatoshi Uehara, Jiawei Huang, and Nan Jiang. Minimax weight and Q-function learning for off-policy evaluation. In International Conference on Machine Learning, pages 9659–9668. PMLR, 2020.
[47] Nathan Kallus and Masatoshi Uehara. Double reinforcement learning for efficient off-policy evaluation in Markov decision processes. J. Mach. Learn. Res., 21:1–63, 2020.
[48] Qing Wang, Jiechao Xiong, Lei Han, Peng Sun, Han Liu, and Tong Zhang. Exponentially weighted imitation learning for batched historical data. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N.

Cesa-Bianchi, and R. Garnett, editors, Advances in Neural Information Processing Systems, volume 31. Curran Associates, Inc., 2018.

[49] Xue Bin Peng, Aviral Kumar, Grace Zhang, and Sergey Levine. Advantage-weighted regression: Simple and scalable off-policy reinforcement learning. arXiv preprint arXiv:1910.00177, 2019.

[50] Xinyue Chen, Zijian Zhou, Zheng Wang, Che Wang, Yanqiu Wu, and Keith Ross. BAIL: Best-action imitation learning for batch deep reinforcement learning. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, Advances in Neural Information Processing Systems, volume 33, pages 18353–18363. Curran Associates, Inc., 2020.

[51] Paria Rashidinejad, Banghua Zhu, Cong Ma, Jiantao Jiao, and Stuart Russell. Bridging offline reinforcement learning and imitation learning: A tale of pessimism. Advances in Neural Information Processing Systems, 34:11702–11716, 2021.

[52] Richard S Sutton. Dyna, an integrated architecture for learning, planning, and reacting. ACM Sigart Bulletin, 2(4):160–163, 1991.

[53] Thomas M Moerland, Joost Broekens, and Catholijn M Jonker. Model-based reinforcement learning: A survey. arXiv preprint arXiv:2006.16712, 2020.

[54] Gen Li, Yuting Wei, Yuejie Chi, Yuantao Gu, and Yuxin Chen. Breaking the sample size barrier in model-based reinforcement learning with a generative model. arXiv preprint arXiv:2005.12900, 2020.

[55] Anurag Ajay, Aviral Kumar, Pulkit Agrawal, Sergey Levine, and Ofir Nachum. OPAL: Offline primitive discovery for accelerating offline reinforcement learning. In International Conference on Learning Representations, 2021.

[56] Romain Laroche, Paul Trichelair, and Remi Tachet Des Combes. Safe policy improvement with baseline bootstrapping. In International Conference on Machine Learning, pages 3652–3661. PMLR, 2019.

[57] Tianhe Yu, Garrett Thomas, Lantao Yu, Stefano Ermon, James Y Zou, Sergey Levine, Chelsea Finn, and Tengyu Ma. MOPO: Model-based offline policy optimization. Advances in Neural Information Processing Systems, 33:14129–14142, 2020.

[58] Byung-Jun Lee, Jongmin Lee, and Kee-Eung Kim. Representation balancing offline model-based reinforcement learning. In International Conference on Learning Representations, 2021.

[59] Michael Lutter, Johannes Silberbauer, Joe Watson, and Jan Peters. Differentiable physics models for real-world offline model-based reinforcement learning. In 2021 IEEE International Conference on Robotics and Automation (ICRA), pages 4163–4170, 2021.

[60] Aayam Kumar Shrestha, Stefan Lee, Prasad Tadepalli, and Alan Fern. DeepAveragers: Offline reinforcement learning by solving derived non-parametric MDPs. In International Conference on Learning Representations, 2021.

[61] Rafael Rafailov, Tianhe Yu, Aravind Rajeswaran, and Chelsea Finn. Offline reinforcement learning from images with latent space models. In Ali Jadbabaie, John Lygeros, George J. Pappas, Pablo A.nbsp;Parrilo, Benjamin Recht, Claire J. Tomlin, and Melanie N. Zeilinger, editors, Proceedings of the 3rd Conference on Learning for Dynamics and Control, volume 144 of Proceedings of Machine Learning Research, pages 1154–1168. PMLR, 07 – 08 June 2021.

[62] Anish Agarwal, Abdullah Alomar, Varkey Alumootil, Devavrat Shah, Dennis Shen, Zhi Xu, and Cindy Yang. PerSim: Data-efficient offline reinforcement learning with heterogeneous agents via personalized simulators. Advances in Neural Information Processing Systems, 34:18564–18576, 2021.

[63] Phillip Swazinna, Steffen Udluft, and Thomas Runkler. Overcoming model bias for robust offline deep reinforcement learning. Engineering Applications of Artificial Intelligence, 104:104366, 2021.

[64] Guy Tennenholtz, Nir Baram, and Shie Mannor. GELATO: Geometrically enriched latent model for offline reinforcement learning. arXiv preprint arXiv:2102.11327, 2021.
[65] Xianyuan Zhan, Haoran Xu, Yue Zhang, Xiangyu Zhu, Honglei Yin, and Yu Zheng. Deepthermal: Combustion optimization for thermal power generating units using offline reinforcement learning. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 36, pages 4680–4688, 2022.
[66] Tianhe Yu, Aviral Kumar, Rafael Rafailov, Aravind Rajeswaran, Sergey Levine, and Chelsea Finn. COMBO: Conservative offline model-based policy optimization. Advances in neural information processing systems, 34:28954–28967, 2021.
[67] Julian Schrittwieser, Thomas K Hubert, Amol Mandhane, Mohammadamin Barekatain, Ioannis Antonoglou, and David Silver. Online and offline reinforcement learning by planning with a learned model. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, Advances in Neural Information Processing Systems, 2021.
[68] Fan-Ming Luo, Tian Xu, Xingchen Cao, and Yang Yu. Reward-consistent dynamics models are strongly generalizable for offline reinforcement learning. In The Twelfth International Conference on Learning Representations, 2023.
[69] Wenxuan Zhou, Sujay Bajracharya, and David Held. PLAS: Latent action space for offline reinforcement learning. In Jens Kober, Fabio Ramos, and Claire Tomlin, editors, Proceedings of the 2020 Conference on Robot Learning, volume 155 of Proceedings of Machine Learning Research, pages 1719–1735. PMLR, 16–18 Nov 2021.
[70] Ziyu Wang, Alexander Novikov, Konrad Zolna, Josh S Merel, Jost Tobias Springenberg, Scott E Reed, Bobak Shahriari, Noah Siegel, Caglar Gulcehre, Nicolas Heess, et al. Critic regularized regression. Advances in Neural Information Processing Systems, 33:7768–7778, 2020.
[71] Rishabh Agarwal, Dale Schuurmans, and Mohammad Norouzi. An optimistic perspective on offline reinforcement learning. In International Conference on Machine Learning, pages 104–114. PMLR, 2020.
[72] Qiang He, Xinwen Hou, and Yu Liu. POPO: Pessimistic offline policy optimization. In ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 4008–4012. IEEE, 2022.
[73] Seyed Kamyar Seyed Ghasemipour, Dale Schuurmans, and Shixiang Shane Gu. EMaQ: Expected-Max Q-learning operator for simple yet effective offline and online RL. In International Conference on Machine Learning, pages 3682–3691. PMLR, 2021.
[74] Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. Conservative Qlearning for offline reinforcement learning. Advances in Neural Information Processing Systems, 33:1179–1191, 2020.
[75] Hua Wei, Deheng Ye, Zhao Liu, Hao Wu, Bo Yuan, Qiang Fu, Wei Yang, and Zhenhui Li. Boosting offline reinforcement learning with residual generative modeling. In Proceedings of the 30th International Joint Conference on Artificial Intelligence (IJCAI), pages 3574–3580, 2021.
[76] Jinning Li, Chen Tang, Masayoshi Tomizuka, and Wei Zhan. Dealing with the unknown: Pessimistic offline reinforcement learning. In 5th Annual Conference on Robot Learning, 2021.
[77] Ilya Kostrikov, Rob Fergus, Jonathan Tompson, and Ofir Nachum. Offline reinforcement learning with Fisher divergence critic regularization. In Marina Meila and Tong Zhang, editors, Proceedings of the 38th International Conference on Machine Learning, volume 139 of Proceedings of Machine Learning Research, pages 5774–5783. PMLR, 18–24 Jul 2021.
[78] Hongchang Zhang, Jianzhun Shao, Yuhang Jiang, Shuncheng He, and Xiangyang Ji. Reducing conservativeness oriented offline reinforcement learning. arXiv preprint arXiv:2103.00098, 2021.

[79] Ofir Nachum, Bo Dai, Ilya Kostrikov, Yinlam Chow, Lihong Li, and Dale Schuurmans. AlgaeDICE: Policy gradient from arbitrary experience. arXiv preprint arXiv:1912.02074, 2019.
[80] Yifan Wu, George Tucker, and Ofir Nachum. Behavior regularized offline reinforcement learning. arXiv preprint arXiv:1911.11361, 2019.
[81] Natasha Jaques, Asma Ghandeharioun, Judy Hanwen Shen, Craig Ferguson, Agata Lapedriza, Noah Jones, Shixiang Gu, and Rosalind Picard. Way off-policy batch deep reinforcement learning of implicit human preferences in dialog. arXiv preprint arXiv:1907.00456, 2019.
[82] Noah Siegel, Jost Tobias Springenberg, Felix Berkenkamp, Abbas Abdolmaleki, Michael Neunert, Thomas Lampe, Roland Hafner, Nicolas Heess, and Martin Riedmiller. Keep doing what worked: Behavior modelling priors for offline reinforcement learning. In International Conference on Learning Representations, 2020.
[83] Jacob Buckman, Carles Gelada, and Marc G Bellemare. The importance of pessimism in fixed-dataset policy optimization. In International Conference on Learning Representations, 2021.
[84] Nuria Armengol Urpı, Sebastian Curi, and Andreas Krause. Risk-averse offline reinforcement learning. In International Conference on Learning Representations, 2021.
[85] Robert Dadashi, Shideh Rezaeifar, Nino Vieillard, L´eonard Hussenot, Olivier Pietquin, and Matthieu Geist. Offline reinforcement learning with pseudometric learning. In Marina Meila and Tong Zhang, editors, Proceedings of the 38th International Conference on Machine Learning, volume 139 of Proceedings of Machine Learning Research, pages 2307–2318. PMLR, 18–24 Jul 2021.
[86] Yue Wu, Shuangfei Zhai, Nitish Srivastava, Joshua M Susskind, Jian Zhang, Ruslan Salakhutdinov, and Hanlin Goh. Uncertainty weighted actor-critic for offline reinforcement learning. In Marina Meila and Tong Zhang, editors, Proceedings of the 38th International Conference on Machine Learning, volume 139 of Proceedings of Machine Learning Research, pages 11319–11328. PMLR, 18–24 Jul 2021.
[87] Yiqin Yang, Xiaoteng Ma, Chenghao Li, Zewu Zheng, Qiyuan Zhang, Gao Huang, Jun Yang, and Qianchuan Zhao. Believe what you see: Implicit constraint approach for offline multi-agent reinforcement learning. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, Advances in Neural Information Processing Systems, 2021.
[88] Jordi Smit, Canmanie T Ponnambalam, Matthijs TJ Spaan, and Frans A Oliehoek. PEBL: Pessimistic ensembles for offline deep reinforcement learning. In Robust and Reliable Autonomy in the Wild Workshop at the 30th International Joint Conference of Artificial Intelligence, 2021.
[89] Scott Fujimoto and Shixiang Shane Gu. A minimalist approach to offline reinforcement learning. Advances in neural information processing systems, 34:20132–20145, 2021.
[90] Shideh Rezaeifar, Robert Dadashi, Nino Vieillard, L´eonard Hussenot, Olivier Bachem, Olivier Pietquin, and Matthieu Geist. Offline reinforcement learning as anti-exploration. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 36, pages 8106–8114, 2022.
[91] Jinzhu Luo and Qi Zhang. Exploiting mdp symmetries for offline reinforcement learning. In CoRL 2023 Workshop on Learning Effective Abstractions for Planning (LEAP), 2023.
[92] Samarth Sinha, Ajay Mandlekar, and Animesh Garg. S4RL: Surprisingly simple self-supervision for offline reinforcement learning in robotics. In Aleksandra Faust, David Hsu, and Gerhard Neumann, editors, Proceedings of the 5th Conference on Robot Learning, volume 164 of Proceedings of Machine Learning Research, pages 907–917. PMLR, 08–11 Nov 2022.
[93] Longyang Huang, Botao Dong, and Weidong Zhang. Efficient offline reinforcement learning with relaxed conservatism. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2024.
[94] Ruoqi Zhang, Ziwei Luo, Jens Sj¨olund, Thomas B Sch¨on, and Per Mattsson. Entropy-regularized

diffusion policy with q-ensembles for offline reinforcement learning. arXiv preprint arXiv:2402.04080, 2024.
[95] Junghyuk Yeom, Yonghyeon Jo, Jungmo Kim, Sanghyeon Lee, and Seungyul Han. Exclusively penalized q-learning for offline reinforcement learning. arXiv preprint arXiv:2405.14082, 2024.
[96] Tengyang Xie, Ching-An Cheng, Nan Jiang, Paul Mineiro, and Alekh Agarwal. Bellman-consistent pessimism for offline reinforcement learning. Advances in neural information processing systems, 34:6683–6694, 2021.
[97] Tengyang Xie, Nan Jiang, Huan Wang, Caiming Xiong, and Yu Bai. Policy finetuning: Bridging sample-efficient offline and online reinforcement learning. Advances in neural information processing systems, 34:27395–27407, 2021.
[98] Ming Yin and Yu-Xiang Wang. Optimal uniform OPE and model-based offline reinforcement learning in time-homogeneous, reward-free and task-agnostic settings. Advances in neural information processing systems, 34:12890–12903, 2021.
[99] Tongzheng Ren, Jialian Li, Bo Dai, Simon S Du, and Sujay Sanghavi. Nearly horizon free offline reinforcement learning. Advances in neural information processing systems, 34:15621–15634, 2021.
[100] Ming Yin, Yu Bai, and Yu-Xiang Wang. Near-optimal provable uniform convergence in offline policy evaluation for reinforcement learning. In Arindam Banerjee and Kenji Fukumizu, editors, Proceedings of The 24th International Conference on Artificial Intelligence and Statistics, volume 130 of Proceedings of Machine Learning Research, pages 1567–1575. PMLR, 13–15 Apr 2021.
[101] Lin Chen, Bruno Scherrer, and Peter L Bartlett. Infinite-horizon offline reinforcement learning with linear function approximation: Curse of dimensionality and algorithm. arXiv preprint arXiv:2103.09847, 2021.
[102] Thanh Nguyen-Tang, Sunil Gupta, Hung Tran-The, and Svetha Venkatesh. Sample complexity of offline reinforcement learning with deep ReLU networks. arXiv preprint arXiv:2103.06671, 2021.
[103] Yichun Hu, Nathan Kallus, and Masatoshi Uehara. Fast rates for the regret of offline reinforcement learning. In Mikhail Belkin and Samory Kpotufe, editors, Proceedings of Thirty Fourth Conference on Learning Theory, volume 134 of Proceedings of Machine Learning Research, pages 2462–2462. PMLR, 15–19 Aug 2021.
[104] Ming Yin, Yu Bai, and Yu-Xiang Wang. Near-optimal offline reinforcement learning via double variance reduction. Advances in neural information processing systems, 34:7677–7688, 2021.
[105] Ruosong Wang, Dean Foster, and Sham M. Kakade. What are the statistical limits of offline RL with linear function approximation? In International Conference on Learning Representations, 2021.
[106] Andrea Zanette. Exponential lower bounds for batch reinforcement learning: Batch RL can be exponentially harder than online rl. In International Conference on Machine Learning, pages 12287–12297. PMLR, 2021.
[107] Xuezhou Zhang, Yiding Chen, Xiaojin Zhu, and Wen Sun. Corruption-robust offline reinforcement learning. In Gustau Camps-Valls, Francisco J. R. Ruiz, and Isabel Valera, editors, Proceedings of The 25th International Conference on Artificial Intelligence and Statistics, volume 151 of Proceedings of Machine Learning Research, pages 5757–5773. PMLR, 28–30 Mar 2022.
[108] Wenhao Zhan, Baihe Huang, Audrey Huang, Nan Jiang, and Jason Lee. Offline reinforcement learning with realizability and single-policy concentrability. In Po-Ling Loh and Maxim Raginsky, editors, Proceedings of Thirty Fifth Conference on Learning Theory, volume 178 of Proceedings of Machine Learning Research, pages 2730–2775. PMLR, 02–05 Jul 2022.

[109] Christian Varschen and Peter Wagner. Mikroskopische modellierung der ersonenverkehrsnachfrage auf basis von zeitverwendungstageb¨uchern. Integrierte Mikro-Simulation von Raum-und Verkehrsentwicklung. Theorie, Konzepte, Modelle, Praxis, 81:63–69, 2006.
[110] Silas C Lobo, Stefan Neumeier, Evelio MG Fernandez, and Christian Facchi. Intas–the ingolstadt traffic scenario for sumo. arXiv preprint arXiv:2011.11995, 2020.
[111] Mengzhe Ruan, Guangfeng Yan, Yuanzhang Xiao, Linqi Song, andWeitao Xu. Adaptive top-k in SGD for communication-efficient distributed learning in multi-robot collaboration. IEEE Journal of Selected Topics in Signal Processing, 18(3):487–501, 2024.
[112] Rishabh Agarwal, Max Schwarzer, Pablo Samuel Castro, Aaron C Courville, and Marc Bellemare. Deep reinforcement learning at the edge of the statistical precipice. Advances in neural information processing systems, 34:29304–29320, 2021.
[113] Takuma Seno and Michita Imai. d3rlpy: An offline deep reinforcement learning library. Journal of Machine Learning Research, 23(315):1–20, 2022.
[114] Denis Tarasov, Alexander Nikulin, Dmitry Akimov, Vladislav Kurenkov, and Sergey Kolesnikov. CORL: Research-oriented deep offline reinforcement learning library. In 3rd Offline RL Workshop: Offline RL as a ”Launchpad”, 2022.
[115] Antonin Raffin. RL Baselines3 Zoo. https://github.com/DLR-RM/rl-baselines3-zoo, 2020.

APPENDIX A HYPERPARAMETER OPTIMIZAITON IN EVALUATION CASES

During hyperparameter optimization, we randomly choose 50 sets of hyperparameters (i.e., learning rate, coefficient of policy entropy, neural network architecture, etc.) for each algorithm. In Table A.1, we show the best average delay and the number of trials that hit the target delay for each algorithm. This provides more detailed statistical information about the hyperparameter optimization.

Table A.1 We show some detailed statistics of the algorithm performance during hyperparameter optimization, in terms of the number of hyperparameter settings under which the algorithm achieves certain thresholds.

4 x 4 Grid – **3** Lanes					
	Best	Delay < 25	Delay < 30	Delay < 35	Delay < 40
ARS	50.0	0	0	0	0
A2C	47.3	0	0	0	0
DQN	**21.2**	**5**	**24**	**26**	**28**
QR-DQN	33.0	0	0	8	13
PPO	22.1	3	20	21	25
TRPO	26.3	0	8	15	17
4 x 4 Grid – **2** Lanes					
	Best	Delay < 25	Delay < 30	Delay < 35	Delay < 40
ARS	135.6	0	0	0	0
A2C	117.8	0	0	0	0
DQN	83.7	3	9	25	34
QR-DQN	100.7	0	0	10	21
PPO	**80.8**	**5**	**15**	**28**	**35**
TRPO	95.6	0	3	12	22

APPENDIX B PYTHON IMPLEMENTATION

The code in this project is built on SUMO-RL [8] and RESCO [9]. We made modification so that we can train the RL algorithms in the framework of RL Baselines3 Zoo [115].

All the code and data will be made publicly available on this GitHub repository:

https://github.com/yuanzhangxiao/sumo-rl-zoo