Online Bootstrap Inference For Policy Evaluation In Reinforcement Learning

Pratik Ramprasad* Yuantong Li† Zhuoran Yang‡ Zhaoran Wang§
Will Wei Sun¶ Guang Cheng‖

June 29, 2022

Abstract

The recent emergence of reinforcement learning (RL) has created a demand for robust statistical inference methods for the parameter estimates computed using these algorithms. Existing methods for inference in online learning are restricted to settings involving independently sampled observations, while inference methods in RL have so far been limited to the batch setting. The bootstrap is a flexible and efficient approach for statistical inference in online learning algorithms, but its efficacy in settings involving Markov noise, such as RL, has yet to be explored. In this paper, we study the use of the online bootstrap method for inference in RL policy evaluation. In particular, we focus on the temporal difference (TD) learning and Gradient TD (GTD) learning algorithms, which are themselves special instances of linear stochastic approximation under Markov noise. The method is shown to be distributionally consistent for statistical inference in policy evaluation, and numerical experiments are included to demonstrate the effectiveness of this algorithm across a range of real RL environments.

Keywords: Asymptotic Normality, Multiplier Bootstrap, Reinforcement Learning, Statistical Inference, Stochastic Approximation

*Department of Statistics, Purdue University. Email: prampras@purdue.edu
†Department of Statistics, UCLA. Email: yuantongli@g.ucla.edu
‡Department of Statistics and Data Science, Yale University. Email: zhuoran.yang@yale.edu
§Department of Industrial Engineering and Management Sciences, Northwestern University. Email: zhaoranwang@gmail.com
¶Krannert School of Management, Purdue University. Email: sun244@purdue.edu. Corresponding author.
‖Department of Statistics, UCLA. Email: guangcheng@stat.ucla.edu

1 Introduction

Reinforcement learning (RL) has achieved phenomenal success in diverse fields, such as robotics (Gu et al., 2017), video games (Mnih et al., 2015), autonomous driving (Sallab et al., 2017), precision medicine (Parekh and Jacobs, 2019), ride-sharing (Xu et al., 2018), and recommendation systems (Chen et al., 2019). A fundamental task in RL is policy evaluation, where the goal is to estimate the value function associated with a given policy from a trajectory of dependent observations. These observations may be sequentially generated either by the same policy (on-policy), or by an unknown behavior policy (off-policy). Standard algorithms used to perform on-policy and off-policy tasks include temporal difference (TD) learning (Sutton, 1988) and gradient temporal difference (GTD) learning (Sutton et al., 2009), respectively. Both are instances of linear stochastic approximation.

While RL has proven to be remarkably successful in various applications, there are still many challenges in real-world systems that prevent it from being applied at scale in practice (Dulac-Arnold et al., 2021). A primary bottleneck in real applications is that environmental interactions are prohibitively expensive. Offline RL (Levine et al., 2020) attempts to address this by training and evaluating multiple policies using pre-existing datasets collected using a single policy, thereby circumventing the need for additional environment interactions. Unfortunately, off-policy value estimates can be challenging in cases where the trained policy is substantially different from the behavior policy, i.e., the policy that was used to collect the data. In such cases, it is important to evaluate policies in an online setting prior to final deployment. Moreover, in many real applications, we are often interested in obtaining not just the point estimate of the value function, but also a measure of the statistical uncertainty associated with the estimate. For example, online randomized experiments, e.g., A/B testing, have been widely conducted by technological/pharmaceutical companies to compare a new product with an old one. Recent studies (Li et al., 2021; Shi et al., 2021, 2022) have used various online updating methods to form sequential testing procedures. In these online evaluation tasks, it is important to quantify the uncertainty of the point estimate for

constructing a valid hypothesis testing. In recommender systems, a new policy is typically tested on a small fraction of user traffic in an online fashion prior to deployment. Providing a confidence interval of the predicted value estimation can help make a recommendation with more confidence because an unstable recommendation can potentially reduce users' trust in the system (Adomavicius and Zhang, 2012; Chen et al., 2019). Similarly, in autonomous driving, policies must be evaluated on real test tracks prior to real-world application and it is often risky to run a policy without a statistically sound estimate of its quality.

Existing inference methods in RL mainly focus on off-policy evaluation using batch updates, which are often computationally inefficient in sequential data scenarios. This motivates the development of online inferential tools for policy evaluation in RL.

Our Contributions

In this work, we present a **fully online** multiplier-type bootstrap algorithm that allows for statistical inference in RL policy evaluation. To the best of our knowledge, this is the first online uncertainty quantification method for sequential decision making under Markovian noise. Since our method is designed for the general framework of stochastic approximation, it applies to both the on-policy and off-policy evaluation tasks, and also to other online learning problems such as stochastic gradient descent (SGD) under Markov noise.

From a theoretical standpoint, our main contribution is to establish the distributional consistency of our bootstrap estimator within the Markovian noise setting. Existing inference methods for online learning (Fang et al., 2018; Chen et al., 2020) have mainly focused on the i.i.d. noise setting, which allows for asymptotic analysis using the martingale properties of the residual noise in the stochastic approximation update. In the present work, the Markovian noise assumption precludes the direct use of martingale central limit theory to characterize the asymptotic properties of our estimator, and necessitates a novel combination of techniques from the Markov Chain Monte-Carlo (MCMC), stochastic approximation, and RL finite-sample analysis literature. Notably, our theoretical results hold under standard assumptions from the RL literature, and do not require any kind of projection step in the

stochastic approximation algorithm, despite the presence of Markovian noise.

Our numerical experiments demonstrate that the algorithm performs on par with the vanilla (offline) bootstrap method across a range of environments, but with substantial savings in terms of data storage and computational cost. This is especially pertinent to computationally demanding algorithms such as deep Q-learning (Mnih et al., 2015), where uncertainty quantification using the standard bootstrap method is often unfeasible due to the massive computational and storage costs involved.

To summarize, our contributions are threefold. First, we develop a fully online inference method for linear stochastic approximation under Markov noise, with applications to both on-policy (TD) and off-policy (GTD) reinforcement learning algorithms. Second, we prove that the confidence intervals constructed using our bootstrap algorithm are asymptotically valid. Third, we demonstrate the efficacy of this method through a number of numerical experiments, including on-policy evaluation with linear TD learning (Sutton and Barto, 2018), deep Q-learning (Mnih et al., 2015), and off-policy evaluation with GTD learning (Sutton et al., 2008, 2009).

Related Work

There has recently been a growth of interest in developing inferential tools for RL and other online learning algorithms such as bandits and SGD. Li et al. (2018) proposed a statistical inference method for M-estimation problems based on fixed step-size SGD. Chen et al. (2020) derived two kinds of estimators for the asymptotic variance of the averaged SGD parameter estimate. Fang et al. (2018) proposed an online estimator for SGD based on randomly perturbing the parameter estimates at each time step. Chen et al. (2021) developed an SGD-based algorithm for online decision making, derived an inverse probability weighted value estimator to estimate the optimal value, and proposed plug-in estimators to estimate the variance of the parameters. While all of these methods are well suited to cases where the data is generated by an i.i.d. sampling procedure, they are unable to account for the underlying dependence structure of the observations generated by RL algorithms.

A closely related field of study is that of dynamic treatment regimes (DTR). It generalizes personalized medicine to time-varying treatment settings where treatments are sequentially adapted to a patient's temporal state. There are similarities to the online learning methods studied here, and recent works have explored the application of RL algorithms such as Q-learning to this domain (Chakraborty and Murphy, 2014), with an emphasis on quantifying the statistical uncertainty associated with their estimators. Ertefaie and Strawderman (2018) develop an estimation procedure for the optimal DTR over an indefinite time period and derive associated large-sample results. Luckett et al. (2019) proposed a new RL method for estimating an optimal treatment regime applicable to the mobile health domain with an infinite time horizon, and established the consistency and asymptotic normality of their estimators under relevant assumptions. These works focus on the estimation of the optimal policy, while ours deals with the inference of the value function for a given policy.

Within the RL and bandit settings, there are a number of recent works that consider the problem of uncertainty quantification of the policy's parameter estimates or value function, focusing mainly on the off-policy setting. Zhang et al. (2021) showed how M-estimation can be extended to provide inferential methods for data collected with bandit algorithms. Kuzborskij et al. (2021) proposed a method for confidence interval estimation based on the Efron-Stein tail inequality within the off-policy contextual bandit setting. Both these methods apply to the bandit setting, where there is no sequential dependence in state transitions. Within the off-policy RL setting, Dai et al. (2020) proposed a method for computing confidence intervals for a target policy's value based on an optimization formulation of the value estimation problem. Jiang and Huang (2020) derive minimax value intervals for off-policy evaluation that satisfy a certain double robustness criterion. However, both papers assume a generative model, where observation tuples are sampled independently from the stationary distribution of the underlying Markov decision process. Shi et al. (2021) proposed an inference method for the state-action value (Q) function via sieve methods to approximate the Q-function. This is an offline method that directly computes the value estimates using

batch updates. On the other hand, ours is a fully online inference method.

A number of recent works in the RL literature (Bhandari et al., 2018; Srikant and Ying, 2019; Xu et al., 2020; Kaledin et al., 2020) have focused on establishing finite sample bounds for the value estimates in TD learning and related algorithms. While these methods provide tight non-asymptotic bounds for the value function under standard assumptions, they do not allow for statistical inference of the estimates. By contrast, the distributional consistency guaranteed by our method allows for the construction of asymptotically exact confidence intervals for the value estimates.

Finally, our online bootstrap algorithm is related to a few recent works that have applied bootstrapping to the bandit and RL settings in various contexts. Wang et al. (2020) proposed a perturbation-based bootstrap exploration method in the bandit setting. Hao et al. (2019) utilized multiplier bootstrap to estimate the upper confidence bound for exploration in the bandit settings. Within the RL setting, White and White (2010) used the moving block bootstrap method to compute confidence intervals for value estimates in continuous Markov decision processes. Hanna et al. (2017) presented two model-based bootstrap methods to compute confidence bounds for off-policy value estimates. Hao et al. (2021) proposed a subsampled bootstrap method for the statistical inference of value estimates computed using the fitted Q-evaluation algorithm. All of these methods require access to the entire dataset in order to carry out the resampling procedure and are therefore only applicable to the batch RL setting. In contrast, the method proposed here computes the bootstrap estimates in an online manner and does not require storage of past observations.

The rest of this paper is organized as follows: In Section 2, we introduce the linear stochastic approximation algorithm and provide some relevant background on the RL algorithms. In Section 3, we present the online bootstrap algorithm. In Section 4, we discuss our theoretical results. In Section 5, we present some numerical simulations that demonstrate the efficacy of the algorithm in various settings, including on-policy TD learning in the FrozenLake RL environment, deep-Q learning in the Atari Pong RL environment, and

off-policy GTD learning in a simulated MDP setting and a real healthcare setting. Finally, in Section 6, we summarize our work and discuss some interesting future directions.

2 Background

2.1 Linear Stochastic Approximation Under Markov Noise

Stochastic approximation is a classic algorithm with a long history in optimization (Robbins and Monro, 1951). In its linear form, the algorithm is designed to solve the equation $\bar{A}\theta = \bar{b}$, where $\bar{A} \in \mathbb{R}^{d\times d}$ and $\bar{b} \in \mathbb{R}^d$ are unknown deterministic quantities, and $\theta \in \Theta \subseteq \mathbb{R}^d$ is the parameter of interest. In the present setting, we are given a sequence of observations of the form $\{(\tilde{A}(X_t), \tilde{b}(X_t))\}_{t\geq 1}$, where $\{X_t\}_{t\geq 1}$ is an ergodic Markov chain with state space $\mathcal{X}$ and stationary distribution μ, and $\tilde{A} : \mathcal{X} \to \mathbb{R}^{d\times d}$ and $\tilde{b} : \mathcal{X} \to \mathbb{R}^d$ are matrix and vector-valued functions defined on the state space $\mathcal{X}$, whose expectations under the stationary distribution μ are $\bar{A}$ and $\bar{b}$, respectively.

The update step for this algorithm is given by

$$\theta_{t+1} = \theta_t + \alpha_{t+1}(\tilde{A}(X_{t+1})\theta_t - \tilde{b}(X_{t+1})), \tag{2.1}$$

where $\theta_t \in \Theta$ is the stochastic approximation iterate i.e., the current estimate of θ, and $\{\alpha_t\}_{t\geq 1}$ is a sequence of polynomially decaying step-sizes, i.e., $\alpha_t = \alpha_0/t^\eta$, for some $\alpha_0 > 0$ and learning rate $\eta \in \left(\frac{1}{2}, 1\right)$.

Our algorithm applies not to the iterate θ_t itself, but to the averaged iterate $\bar{\theta}_t = \frac{1}{t}\sum_{i=1}^t \theta_i$. This averaging scheme is referred to as Polyak-Ruppert averaging, after Ruppert (1988) and Polyak and Juditsky (1992), who established the asymptotic normality of $\bar{\theta}_t$ for strongly convex objective functions under Martingale noise.

2.2 Policy Evaluation In Reinforcement Learning

In this section, we briefly review some background theory on Markov reward processes and policy evaluation, and describe the temporal difference (TD) learning and Gradient TD

(GTD) learning algorithms as instances of the linear stochastic approximation algorithm with Markov noise in (2.1). In addition to these, recent studies (Mou et al., 2021; Durmus et al., 2021; Chen et al., 2021) show that other TD variates like n-step TD and TD(λ) are also special cases of Markovian linear stochastic approximation algorithms, which enables their studies on the finite-sample convergence guarantees of these RL algorithms. Our setting can be viewed an a complementary to the martingale noise setting considered in offline policy evaluation problems (Luckett et al., 2019; Shi et al., 2021). These methods do not allow for an online update but instead consider a batch update or an offline update.

Markov Reward Processes

A Markov Decision Process (MDP) is denoted as $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma)$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ is a finite set of actions, $\mathcal{P}$ is the transition kernel, $\mathcal{R}$ is the reward function, and $\gamma \in (0, 1)$ is a discount factor.

A stationary policy π maps a state $s \in \mathcal{S}$ to a probability space via the distribution $\pi(\cdot|s)$. At time step t, suppose the learner at state $s_t \in \mathcal{S}$, following the policy π, takes the action $a_t \in \mathcal{A}$ with probability $\pi(a_t|s_t)$. Then the transition kernel $\mathcal{P}(s_{t+1}|s_t, a_t)$ determines the probability of being in the next state $s_{t+1} \in \mathcal{S}$ at the next time step, and the reward $r_{t+1} = \mathcal{R}(s_t, a_t, s_{t+1})$, assumed to be bounded by $r_{\max}$, is obtained.

The stationary policy π and the MDP together induce a Markov Reward Process (MRP) $\mathcal{M}^\pi = (\mathcal{M}, \pi)$, with transition kernel $\mathcal{P}^\pi(s'|s) = \sum_{a\in\mathcal{A}} \mathcal{P}(s'|s, a)\pi(a|s)$. Similarly, the expected reward function of the MRP is given by

$$\mathcal{R}^\pi(s) = \sum_{a\in\mathcal{A}} \pi(a|s) \sum_{s'\in\mathcal{S}} \mathcal{P}(s'|s, a)\mathcal{R}(s, a, s').$$

The ergodicity of the Markov chain ensures the existence of a stationary state distribution μ_π for the MRP over the state space under the stationary policy π.

Value Functions

The value function associated with a policy π, denoted as $V^\pi : \mathcal{S} \to \mathbb{R}$, is the discounted sum of expected rewards from starting at a state and following policy π:

$$V^\pi(s) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t \mathcal{R}^\pi(s_t) | s_0 = s\right], \quad s \in \mathcal{S},$$

where the expectation is taken over the set of trajectories generated according to the transition kernel $\mathcal{P}^\pi$.

The value function is the unique solution to the Bellman equation

$$V^\pi(s) = \mathcal{R}^\pi(s) + \gamma \mathbb{E}_{s' \sim \mathcal{P}^\pi(\cdot|s)}\left[v^\pi(s')\right], \quad s \in \mathcal{S}. \tag{2.2}$$

We define the Bellman operator on the space of value functions as

$$T^\pi V(s) = \mathcal{R}^\pi(s) + \gamma \mathbb{E}_{s' \sim \mathcal{P}^\pi(\cdot|s)}[V(s')],$$

for any value function $V : \mathcal{S} \to \mathbb{R}$. Then V^π is the unique fixed point of the operator T^π.

TD Learning With Linear Function Approximation

TD learning is widely used for the estimation of the value function for a given policy. The classic version of this algorithm, now known as tabular TD learning (Sutton, 1988), attempts to compute value estimates for every state in the state space. In modern RL applications, such an approach becomes infeasible, as real-world problems often involve very large state spaces. In those cases, a natural approach is to approximate the value function as $V_\theta(s) = \sum_{i=1}^d \varphi_i(s)\theta_i = \phi(s)^\mathsf{T}\theta$, where $\{\varphi_i : \mathcal{S} \to \mathbb{R}\}_{i=1}^d$ is a set of basis functions, $\phi(s) = [\varphi_1(s), \ldots, \varphi_d(s)]^\mathsf{T}$, and $\theta \in \mathbb{R}^d$ denotes the parameter. Then, given a sequence of observations of the form $\{(s_t, r_{t+1}, s_{t+1})\}_{t \geq 0}$, the linear TD update is given by

$$\theta_{t+1} \leftarrow \theta_t + \alpha_{t+1}\left(\left(\phi(s_t) - \gamma\phi(s_{t+1})^\mathsf{T}\theta_t - r_{t+1}\right)\phi(s_t)\right) \tag{2.3}$$

When the state space $\mathcal{S}$ is finite, the states may be enumerated as $\{1, \ldots, |\mathcal{S}|\}$, with the integer i corresponding to $s^{(i)}$, the ith state in $\mathcal{S}$. We can then express the transition kernel

$\mathcal{P}^\pi$ as the matrix $P^\pi \in \mathbb{R}^{|\mathcal{S}|\times|\mathcal{S}|}$, with $P^\pi_{i,j} = \mathcal{P}^\pi(s^{(j)}|s^{(i)})$, and the expected rewards at each state as the vector $r^\pi \in \mathbb{R}^{|\mathcal{S}|}$, with $r^\pi_i = \mathcal{R}^\pi(s^{(i)})$.

Using this formulation, linear TD learning may be seen as an instance of the linear stochastic approximation update (2.1), solving the linear equation $\bar{A}\theta = \bar{b}$, with

$$\bar{A} = \Phi^\mathsf{T}\Xi(I - \gamma P^\pi)\Phi, \quad \text{and} \quad \bar{b} = \Phi^\mathsf{T}\Xi r^\pi,$$

where $\Phi \in \mathbb{R}^{|\mathcal{S}|\times d}$ denotes the feature matrix, containing the features $\phi(s^{(i)}), 1 \leq i \leq |\mathcal{S}|$ in its rows, and $\Xi \in \mathbb{R}^{|\mathcal{S}|\times|\mathcal{S}|}$ is a diagonal matrix with elements corresponding to the entries of the stationary distribution μ_π.

For samples $X_t = (s_t, r_{t+1}, s_{t+1})$ generated sequentially from the MRP $\mathcal{M}^\pi$, we can run linear stochastic approximation using the quantities

$$\tilde{A}(X_t) = \phi(s_t)\left(\phi(s_t) - \gamma\phi(s_{t+1})\right)^\mathsf{T}, \quad \text{and} \quad \tilde{b}(X_t) = r_{t+1}\phi(s_t).$$

GTD Learning For Off-Policy Evaluation

In off-policy evaluation, the goal is to estimate the value of a target policy π, given a set of observations generated by a behavior policy π_b. This is an important problem in RL, as it enables us to evaluate several policies using data generated by a different behavior policy.

Unlike the on-policy setting, the traditional TD learning is no longer feasible due to its convergence issue in the off-policy setting (Sutton and Barto, 2018). On the other hand, Gradient TD (GTD) algorithms (Sutton et al., 2008, 2009) are guaranteed to converge even in the off-policy setting.

The algorithm uses a form of importance sampling to correct for the discrepancy between the target and behavior policy. This is done by scaling the updates by an importance sampling ratio $\rho_t = \frac{\pi(a_t|s_t)}{\pi_b(a_t|s_t)}$.

There are two variants of this algorithm, GTD1 and GTD2, which seek to minimize the Norm of the Expected TD Update (NEU), and the Mean-Square Projected Bellman Error (MSPBE), respectively. These loss functions have a similar structure, and can be unified as

$$J(\theta) = \|\Phi^\mathsf{T}\Xi(r^\pi + \gamma P^\pi \hat{v}_\theta - \hat{v}_\theta)\|^2_{M^{-1}},$$

where $M \in \mathbb{R}^{d\times d}$ is the identity under NEU, and $M = \Phi^\mathsf{T}\Xi\Phi$ under MSPBE.

In order to minimize this loss function, a pseudo-stochastic gradient method was proposed, involving two simultaneous updates:

$$y_{t+1} = y_t + \alpha_{t+1}(b_t + A_t\theta_t - M_t y_t),$$
$$\theta_{t+1} = \theta_t + \alpha_{t+1}A_t^T y_t,$$

where $b_t = \rho_t r_{t+1}\phi(s_t)$ and $A_t = \rho_t\phi(s_t)\left(\phi(s_t) - \gamma\phi(s_{t+1})\right)^\mathsf{T}$ are unbiased estimators for $b = \Phi^\mathsf{T}\Xi r^\pi$ and $A = \Phi^\mathsf{T}\Xi(I - \gamma P^\pi)\Phi$, respectively. Similarly, M_t is an unbiased estimator for M, with $M_t = I_d$ under NEU and $M_t = \phi(s_t)\phi(s_t)^\mathsf{T}$ under MSPBE.

These two steps can be combined into a single linear stochastic approximation update solving the linear equation $\bar{A}\Theta = \bar{b}$, where

$$\bar{A} = \begin{pmatrix} 0 & -A^\mathsf{T} \\ A & M \end{pmatrix}, \quad \bar{b} = \begin{pmatrix} 0 \\ b \end{pmatrix}, \quad \text{and} \quad \Theta = \begin{pmatrix} \theta \\ y \end{pmatrix}.$$

As in case of TD learning with linear function approximation, for samples generated sequentially under the behavior policy π_b, we denote the observed tuples as $X_t = (s_t, r_{t+1}, s_{t+1})$. Then we can run linear stochastic approximation using the quantities

$$\tilde{A}(X_t) = \begin{pmatrix} 0 & -A_t^\mathsf{T} \\ A_t & M_t \end{pmatrix}, \quad \tilde{b}(X_t) = \begin{pmatrix} 0 \\ b_t \end{pmatrix}, \quad \text{and} \quad \Theta_t = \begin{pmatrix} \theta_t \\ y_t \end{pmatrix}.$$

3 Method

3.1 Online Bootstrap Algorithm

Given a dataset $\{\tilde{A}(X_t), \tilde{b}(X_t)\}_{t\geq 1}$ of sequentially generated observations, the Polyak-Ruppert averaged iterate estimates the parameter θ_* as $\bar{\theta}_t = \frac{1}{t}\sum_{i=1}^t \theta_i$, where θ_i is defined in (2.1). Under the assumptions listed in Section 4, $\bar{\theta}_t$ is a consistent estimator of θ_*, and its distribution is asymptotically Gaussian with mean θ_* and a certain covariance matrix. The estimation of this asymptotic covariance is crucial for statistical inference, but there is presently no straightforward way to estimate it in the presence of Markov noise. This motivates the development of the online bootstrap method.

The online bootstrap method is based on the following perturbed stochastic approximation update, which is performed in parallel to the main update (2.1):

$$\hat{\theta}_{t+1} = \hat{\theta}_t + \alpha_{t+1} W_{t+1} (\tilde{A}(X_{t+1})\hat{\theta}_t - \tilde{b}(X_{t+1})), \tag{3.1}$$

where $\{W_t\}_{t\geq 1}$ is a bounded sequence of i.i.d. random variables with mean 1 and variance 1, with $W_1 < W_{\max}$ a.s. for some finite constant $W_{\max} > 0$. The $\hat{\cdot}$ notation is used here to distinguish the perturbed iterates from those of the standard update (2.1). Let $\bar{\hat{\theta}}_t$ denote the averaged iterate of the sequence $\{\hat{\theta}_t\}_{t\geq 0}$. In Theorem 4.2 of the Section 4, we show that the distributions of $\sqrt{t}(\bar{\theta}_t - \theta_*)$ and $\sqrt{t}(\bar{\hat{\theta}}_t - \bar{\theta}_t)$ are asymptotically equivalent. This is a fundamental result for the validity of our online bootstrap algorithm, as it enables us to conduct inference on the former distribution by using the latter as a proxy. The latter distribution may be approximated by bootstrapping B samples of $\bar{\hat{\theta}}_t - \bar{\theta}_t$. For each $b = 1, \ldots, B$, at time step $t+1$, we update the perturbed SGD iterates $\hat{\theta}_t^{(b)}$ as follows:

$$\begin{aligned}
\hat{\theta}^b_{t+1} &= \hat{\theta}^b_t + \alpha_{t+1} W^b_{t+1} (\tilde{A}(X_{t+1})\hat{\theta}^b_t - \tilde{b}(X_{t+1})), \\
\bar{\hat{\theta}}^b_{t+1} &= \frac{1}{t+1} \sum_{i=1}^{t+1} \hat{\theta}^b_i,
\end{aligned}$$

where W^b_t are i.i.d. random variates with mean one and variance one. The updates can be performed in a fully online manner, as they only rely on the latest available data point X_{t+1}. Furthermore, since all B trajectories of perturbed iterates depend on a single trajectory of the Markov chain $\{X_t\}$, the iterates can be updated in parallel. Intuitively, the bootstrap method proposed here circumvents the issue of higher-order dependence in the SA iterates by enabling us to perform statistical inference using the cross-section of perturbed iterates generated at each time step, as opposed to using the highly dependent iterates generated by a single trajectory. Algorithm 1 presents the entire online update scheme.

3.2 Constructing Confidence Intervals

We use two approaches to construct confidence intervals using the bootstrap empirical distribution - the quantile and the standard error estimators. As the names suggest, these are

Algorithm 1: Online Bootstrap For Linear Stochastic Approximation

Input : Number of bootstrap samples B, Initial step size $\alpha_0 > 0$, Learning rate $\eta \in (\frac{1}{2}, 1)$, Initial estimates $\theta_0 = \hat{\theta}_0^b, b = 1, \ldots, B$.

for $t = 0, 1, 2, \ldots$ **do**
  Observe $\tilde{A}(X_{t+1}), \tilde{b}(X_{t+1})$.
  Compute $\alpha_{t+1} = \alpha_0 \eta^{-(t+1)}$.
  Update $\theta_{t+1} \leftarrow \theta_t + \alpha_{t+1}(\tilde{A}(X_{t+1})\theta_t - \tilde{b}(X_{t+1}))$.
  Update $\bar{\theta}_{t+1} \leftarrow \frac{1}{t+1}(t\bar{\theta}_t + \theta_{t+1})$.
  for $b = 1, 2, \ldots, B$ **do**
    Update $\hat{\theta}_{t+1}^b \leftarrow \hat{\theta}_t^b + \alpha_{t+1} W_{t+1}^b (\tilde{A}(X_{t+1})\hat{\theta}_t^b - \tilde{b}(X_{t+1}))$.
    Update $\bar{\hat{\theta}}_{t+1}^b \leftarrow \frac{1}{t+1}(t\bar{\hat{\theta}}_t^b + \hat{\theta}_{t+1}^b)$.
  end
end

Output: Bootstrap estimates $\left\{\bar{\hat{\theta}}_{t+1}^b\right\}_{b=1}^B$

based on estimating the quantiles and standard errors of the bootstrap errors. Let q_δ denote the δth quantile of the empirical bootstrap distribution $\left\{\bar{\hat{\theta}}_t^{(b)} - \bar{\theta}_t\right\}_{b=1}^B$. Then the $(1-\alpha)$ quantile-based confidence interval for θ is given by $(\bar{\theta}_t + q_{\alpha/2}, \bar{\theta}_t + q_{1-\alpha/2})$.

Similarly, let $\hat{\Sigma}$ denote the sample covariance matrix of the empirical distribution, and let $\hat{\sigma} = \sqrt{\text{diag}(\hat{\Sigma})}$. Then the $(1-\alpha)$ standard error-based confidence interval for θ is given by $(\bar{\theta}_t + z_{\alpha/2}\hat{\sigma}, \bar{\theta}_t + z_{1-\alpha/2}\hat{\sigma})$, where z_α denotes the αth quantile of the standard normal distribution. The validity of the SE confidence interval relies on the consistency of the second moment of $\bar{\theta}_t$. Moment consistency is directly implied by the distributional consistency result of Theorem 4.2 under slightly stronger conditions (Cheng, 2015).

In the task of policy evaluation in RL, we are interested in constructing confidence intervals for the value function. Under linear function approximation, the value estimate corresponding to the policy π, given the averaged iterate $\bar{\theta}_t$, is $v_{\bar{\theta}_t}(s) = \phi(s)^\intercal \bar{\theta}_t$, where $\phi(s)$ denotes the feature mapping for the state $s \in \mathcal{S}$. More generally, we are interested in estimating the value associated with a reference distribution ν over the state space $\mathcal{S}$. In this case, the value estimate for the policy π is given by

$$V_{\bar{\theta}_t}^\pi(\nu) = \int_{s \in \mathcal{S}} \phi(s)^\intercal \bar{\theta}_t \nu(ds).$$

Confidence intervals for $V_{\bar{\theta}_t}^{\pi}(\nu)$ can be constructed using the quantile or standard error estimators. Let q_δ denote the δth quantile of the bootstrapped value estimates $\left\{V_{\tilde{\theta}_t^{(b)}}^{\pi}(\nu) - V_{\bar{\theta}_t}^{\pi}(\nu)\right\}_{b=1}^{B}$. Then the $(1-\alpha)$ quantile-based confidence interval for the value estimate is given by

$$\left(V_{\bar{\theta}_t}^{\pi}(\nu) + q_{\alpha/2}, V_{\bar{\theta}_t}^{\pi}(\nu) + q_{1-\alpha/2}\right).$$

Similarly, let

$$\tilde{\sigma}(\nu) = \left(\int_{s\in\mathcal{S}} \phi(s)\nu(ds)\right)^{\mathsf{T}} \hat{\Sigma} \left(\int_{s\in\mathcal{S}} \phi(s)\nu(ds)\right)$$

denote the sample covariance matrix of the bootstrapped value estimates. Then the $(1-\alpha)$ standard error-based confidence interval for the value estimate is given by

$$\left(V_{\bar{\theta}_t}^{\pi}(\nu) + z_{\alpha/2}\tilde{\sigma}(\nu), V_{\bar{\theta}_t}^{\pi}(\nu) + z_{1-\alpha/2}\tilde{\sigma}(\nu)\right).$$

4 Distributional Consistency Result

4.1 Asymptotic Theory For Linear Stochastic Approximation Under Markov Noise

In this section, we list our assumptions and state some relevant results pertaining to the asymptotic theory of linear stochastic approximation.

(A1). *The Markov chain $\{X_t\}$ is uniformly ergodic, with transition kernel $\mathcal{P}$ and unique stationary distribution μ.*

Assumption (A1) is standard in the RL literature for the policy evaluation setting, especially in recent works on the finite-sample analysis of policy evaluation algorithms (Bhandari et al., 2018; Srikant and Ying, 2019; Xu et al., 2020). It always holds for irreducible, aperiodic Markov chains (Levin et al., 2017).

As a direct consequence of (A1), there exist constants $M > 0$ and $\kappa \in (0,1)$ such that

$$\sup_{x\in\mathcal{X}} \left\|\mathcal{P}^t(x,\cdot) - \mu\right\| \leq M\kappa^t, \tag{4.1}$$

where $\mathcal{P}^t(x, \cdot)$ denotes the t-step transition kernel, starting from state x (see e.g. Meyn and Tweedie (2009); Douc et al. (2018)). Equation (4.1) characterizes the rate at which the Markov chain $\{X_t\}$ approaches its stationary distribution μ, when starting from an arbitrary initial distribution.

Next, we define conditions on the noisy observations:

(A2). *(i) There exists a matrix $\bar{A}$ and a vector $\bar{b}$ such that*

$$\bar{A} = \lim_{t\to\infty} \mathbb{E}[\tilde{A}(X_t)], \quad \textit{and} \quad \bar{b} = \lim_{t\to\infty} \mathbb{E}[\tilde{b}(X_t)].$$

(ii) The matrix $\bar{A}$ is full rank and Hurwitz, i.e., all its eigenvalues have strictly negative real parts.

(iii) There exist constants $A_{\max}$ and $b_{\max}$ such that

$$\sup_{x\in\mathcal{X}} \left\|\tilde{A}(x)\right\|_F \leq A_{\max}, \quad \sup_{x\in\mathcal{X}} \left\|\tilde{b}(x)\right\|_2 \leq b_{\max}.$$

The conditions listed in (A2) are imposed in order to ensure convergence of the sequence of iterates $\{\theta_t\}$. (A2)(i) is self-explanatory, while (A2)(ii) is a standard assumption in the stochastic approximation literature that ensures the stability of the algorithm (Polyak and Juditsky, 1992; Srikant and Ying, 2019; Chen et al., 2020). It is generally considered a reasonable assumption in the RL setting, both for TD learning (Bhandari et al., 2018; Hu and Syed, 2019) and for GTD learning (Gupta et al., 2019). (A2)(iii) controls the behavior of the Markov noise. In the RL setting, it holds whenever the feature maps ϕ and the reward function $\mathcal{R}$ are bounded (Sutton and Barto, 2018). By (A2)(i) and (A2)(iii), we also have that $\left\|\bar{A}\right\|_F \leq A_{\max}$ and $\left\|\bar{b}\right\|_2 \leq b_{\max}$.

By (A2), there exists a unique solution $\theta_* \in \Theta$ to the linear equation $\bar{A}\theta = \bar{b}$. Furthermore, we can now write (2.1) as

$$\theta_{t+1} = \theta_t + \alpha_{t+1}(\bar{A}\theta_t - \bar{b}) + \alpha_{t+1}\epsilon_{t+1}, \tag{4.2}$$

where $\epsilon_{t+1} = (\tilde{A}(X_{t+1}) - \bar{A})\theta_t - (\tilde{b}(X_{t+1}) - \bar{b})$ is a residual noise term.

Our final assumption has to do with the step sizes $\{\alpha_t\}$:

(A3). *The step sizes are of the form* $\alpha_t = \alpha_0/t^\eta$, $t \geq 1$, *where* $\alpha_0 > 0$ *and the learning rate* $\eta \in (\frac{1}{2}, 1)$.

Polynomially decaying step sizes are standard in the case of iterate averaging (Polyak and Juditsky, 1992).

Under these conditions, we have the following almost sure rate of convergence result for the update (4.2):

Proposition 4.1. *Suppose that (A1)-(A3) hold. Let* $\eta \in (1/2, 1)$ *be defined as in (A3), and let* $\gamma \in (0, \eta - 1/2)$. *Then the iterates of update (4.2) satisfy* $\|\theta_t - \theta_*\|_2 = o(t^{-\gamma})$, *a.s.*

This result establishes the consistency of the LSA iterate θ_t in the Markov noise setting. Additionally, it ensures that the iterates are bounded within a compact set without the need for a projection scheme. A proof of the result is provided in the supplementary section.

By Lemma A.5 of Liang (2010), we can split the noise term in (4.2) into three parts as

$$\epsilon_t = e_t + \nu_t + \zeta_t, \tag{4.3}$$

where e_t is a martingale difference sequence, i.e., $\mathbb{E}[e_t|\mathcal{F}_{t-1}] = 0$, where $\mathcal{F}_t$ is the natural filtration associated with the Markov chain $\{X_t\}$, while ν_t and ζ_t are decaying residual noise terms.

Proposition 4.2. *Assume conditions (A1)-(A3) hold. Then*

$$\sqrt{t}(\bar{\theta}_t - \theta_*) \implies \mathcal{N}(0, \bar{A}^{-1}Q(\bar{A}^{-1})^\mathsf{T}),$$

where $Q = \lim_{t\to\infty} \mathbb{E}[e_t e_t^\mathsf{T}]$, *with* e_t *defined as in (4.3).*

Proposition 4.2 establishes a central limit theorem for the averaged iterate and provides an explicit form for the asymptotic variance. Under i.i.d. noise, the asymptotic variance may be estimated using a plug-in estimator (Chen et al., 2021). However, in the present setting, we have $Q = \lim_{t\to\infty} \mathbb{E}[e_t e_t^\mathsf{T}]$, where e_t is the martingale component of the Markov noise term ϵ_t, as per the decomposition (4.3). To the best of our knowledge, there is no existing

method to estimate e_t by separating it from the other components of ϵ_t. Our online bootstrap algorithm provides an efficient and theoretically sound way to estimate the distribution of $\bar{\theta}_t$. The properties of its key component, the perturbed linear stochastic approximation, are discussed in the following section.

4.2 Perturbed Linear Stochastic Approximation

We now study the asymptotic behavior of the perturbed linear stochastic approximation update (3.1). As in the standard case, the mean field here is $h(\theta) = \bar{A}\theta - \bar{b}$, while the observations are of the form $H(\theta, X_{t+1}) = W_{t+1}(\tilde{A}(X_{t+1}\theta - \tilde{b}(X_{t+1}))$. By the independence and boundedness of W_t, $H(\theta, X_{t+1})$ is an unbiased estimator of $h(\theta)$ under the stationary distribution of X_t.

We may rewrite (3.1) in terms analogous to (4.2), as follows:

$$\hat{\theta}_{t+1} = \hat{\theta}_t + \alpha_{t+1}(\bar{A}\hat{\theta}_t - \bar{b}) + \alpha_{t+1}\hat{\epsilon}_{t+1}, \tag{4.4}$$

where $\hat{\epsilon}_{t+1} = (W_{t+1}\tilde{A}(X_{t+1}) - \bar{A})\hat{\theta}_t - (W_{t+1}\tilde{b}(X_{t+1}) - \bar{b})$ is the noise term.

Having established the almost sure rate of convergence for the standard LSA update (4.2), the result may be straightforwardly extended to the perturbed update (4.4):

Proposition 4.3. *Suppose that (A1)-(A3) hold. Let $\eta \in (1/2, 1)$ be defined as in (A3), and let $\gamma \in (0, \eta - 1/2)$. Then the iterates of update (4.4) satisfy $\left\|\hat{\theta}_t - \theta_*\right\|_2 = o(t^{-\gamma})$, a.s.*

Next, we establish the theoretical validity of the online bootstrap algorithm of Section 3. For this, we need the following lemma:

Lemma 4.1. *Assume that (A1)-(A3) hold. Then*

$$\sqrt{t}(\bar{\hat{\theta}}_t - \theta_*) = -\frac{1}{\sqrt{t}}\bar{A}^{-1}\sum_{i=1}^{t} W_{i+1}(\tilde{A}(X_{i+1})\theta_* - \tilde{b}(X_{i+1})) + o_p(1). \tag{4.5}$$

The proof is provided in the supplementary section. Note that Lemma 4.1 also holds for the update (4.2), since (4.4) reduces to (4.2) when $W_i \equiv 1$ for all i. Hence we have

$$\sqrt{t}(\bar{\theta}_t - \theta_*) = -\frac{1}{\sqrt{t}}\bar{A}^{-1}\sum_{i=1}^{t}(\tilde{A}(X_{i+1})\theta_* - \tilde{b}(X_{i+1})) + o_p(1). \tag{4.6}$$

Subtracting (4.6) from (4.5), we get

$$\sqrt{t}(\bar{\hat{\theta}}_t - \bar{\theta}_t) = -\frac{1}{\sqrt{t}}\bar{A}^{-1}\sum_{i=1}^{t}(W_{i+1}-1)(\tilde{A}(X_{i+1})\theta_* - \tilde{b}(X_{i+1})) + o_p(1). \tag{4.7}$$

The final step is to establish the distributional consistency of the online bootstrap estimator in the Kolmogorov metric. Suppose that the data $\mathcal{D}$ is generated under the probability space $\{\mathcal{X}, \mathcal{A}, \mathbb{P}_{\mathcal{D}}\}$, while the bootstrap weights $\mathcal{W} = \{W_i\}_{i=1}^t$ are generated under an independent probability space $(\Omega, \mathcal{B}, \mathbb{P}_{\mathcal{W}})$. Let $\mathbb{P}_{\mathcal{W}|\mathcal{D}}$ denote the conditional distribution given the observed data $\mathcal{D}$.

Theorem 4.2. *Assume that (A1)-(A3) hold. Then, as $B \to \infty$ and $t \to \infty$, we have*

$$\sup_{v\in\mathbb{R}^d}\left|\mathbb{P}_{\mathcal{W}|\mathcal{D}}\left(\sqrt{t}(\bar{\hat{\theta}}_t - \bar{\theta}_t) \le v\right) - \mathbb{P}_{\mathcal{D}}\left(\sqrt{t}(\bar{\theta}_t - \theta_*) \le v\right)\right| \to 0, \textit{ in probability.} \tag{4.8}$$

The proof is provided in the supplementary section. We note that the above result applies to the sequence of probability measures $\mathbb{P}_{W|\mathcal{D}}$, where the datasize and the number of bootstrap samples grows to infinity. In practice, there is a bootstrap Monte-Carlo error associated with re-sampling from a finite number of bootstrap samples B. By choosing B adequately large, the bootstrap Monte-Carlo error is generally ignored (DasGupta, 2008).

Theorem 4.2 establishes the theoretical foundation for using our online bootstrap estimator for statistical inference. The distributional consistency of the bootstrap estimator in terms of the Kolmogorov metric (4.8) enables us to construct asymptotically exact confidence intervals for the estimator $\hat{\theta}$ or functions of the estimator, such as the value estimate in TD learning under linear function approximation.

5 Experiments

In this section, we evaluate the performance of the online bootstrap algorithm through numerical simulations across various settings. We construct two types of confidence intervals using the online bootstrap algorithm - namely, the quantile estimator and the standard error estimator, as defined in the previous section. All confidence intervals were constructed to

have a coverage of 95%. In all cases, we set the number of bootstrap samples B to 200, the learning rate η to 3/4, and the initial parameter θ_0 to the zero vector. The random variates W_t are sampled from the uniform distribution over the interval $(1-1/\sqrt{3}, 1+1/\sqrt{3})$, so they are bounded and have mean 1, variance 1.

While our results apply most naturally to the infinite-horizon (continual) setting due to our assumption of ergodicity and the existence of a unique stationary distribution, they can be applied to the finite-horizon (episodic) setting in a straightforward manner. To accomplish this, we simply concatenate the individual episodes to form a trajectory of infinite length, and perform inference on the parameters with respect to this infinite-horizon embedding of the MDP. Similar approach has also been applied in the experiments of other RL literature (Dai et al., 2020; Xu et al., 2020). The theoretical validity of such a concatenation procedure has been studied by Bojun (2020).

The main benchmark we use to measure our algorithm's performance is the vanilla bootstrap, which is an offline method that requires the entire batch of samples for computation. Bootstrap-based inference is highly versatile with regard to the choice of policy evaluation algorithm, and is known to provide better second-order accuracy even when the asymptotic distribution is available (Hall, 1992). It has already been studied in the RL literature in various contexts (White and White, 2010; Hanna et al., 2017; Hao et al., 2021), and naturally allows for a like-for-like comparison with our method. Therefore, we opt to use the vanilla bootstrap as our primary comparison method. In order to ensure a fair comparison, we use the same number of bootstrap samples, and within each sample, we use the same number of re-sampled observations as the size of the original dataset.

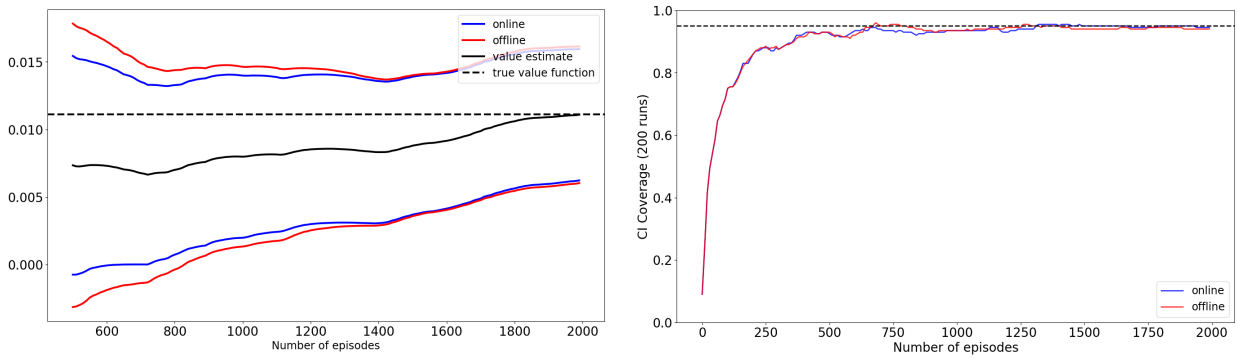
5.1 On-Policy Value Inference For FrozenLake RL Environment

Next, we consider the Frozenlake environment from OpenAI gym (Brockman et al., 2016). Here the RL agent controls the movement of a character in an 8×8 grid world. The starting point is the first tile of the grid, and the goal is to reach the end tile of the grid. The rest of

the tiles are either walkable or absorbing states. A reward of 1 is awarded if the character reaches the target tile, and the reward for any other state transition is 0.

We use linear TD learning to estimate the value function associated with a near-optimal policy trained using Q-learning (Sutton and Barto, 2018). For the compared vanilla (offline) bootstrap method, we choose to resample the observations by episodes, rather than by sample transitions, as suggested by Hao et al. (2021), as the sample transitions may fail to capture the sequential dependence in state transitions.

Figure 1a shows an example of the confidence intervals generated for the value estimate of the initial state, using both the online and offline bootstrap methods. The true value function for that state, computed analytically using the transition probability matrix, is included for reference. As in the previous experiment, we also examine the empirical coverage probabilities, in Figure 1b. Both methods are seen to achieve the nominal coverage of 95% within approximately 1000 episodes.



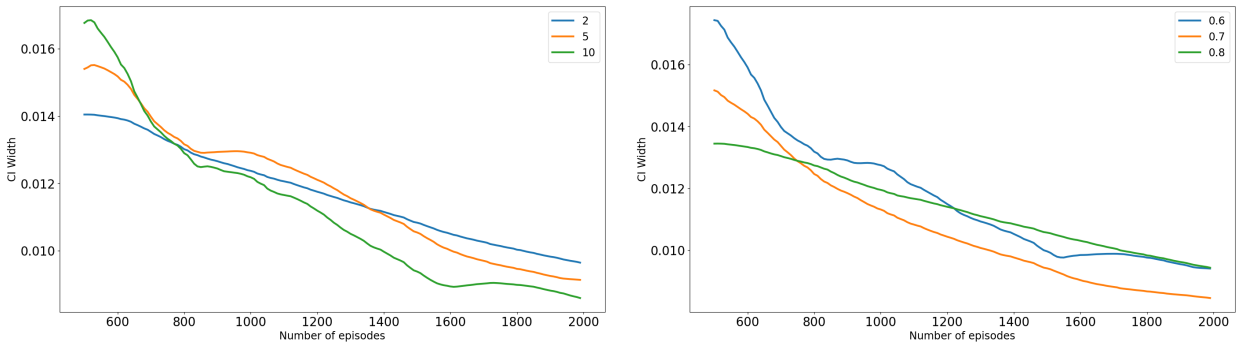
(a) CI Example

(b) Empirical Coverage Probabilities

Figure 1: Statistical inference in the Frozen Lake RL environment: Figure 1a shows an example trajectory for the CIs generated using the online and offline bootstrap methods. The value estimates are for the initial state; the true value for that state is included for comparison. Figure 1b shows the empirical coverage probabilities for the value estimates of the initial state, based on 200 repeated experiments using the online and offline bootstrap methods, with 2000 episodes per run.

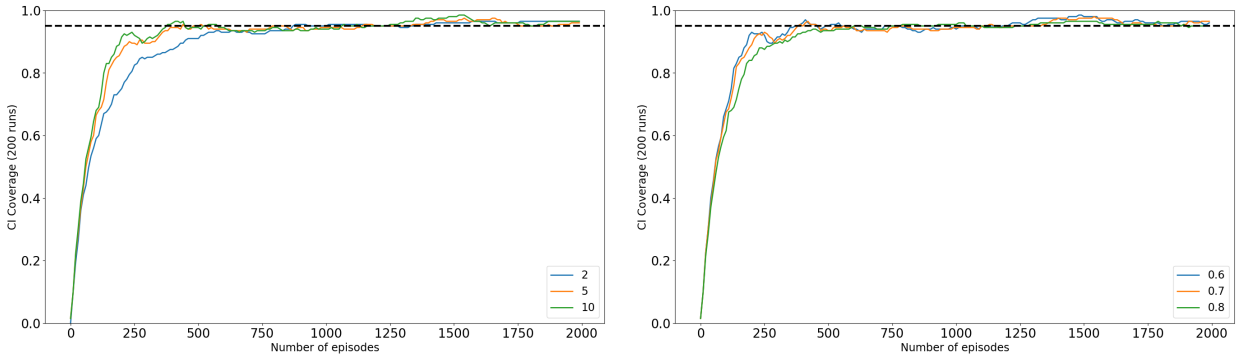
Figure 2 shows the sensitivity of the widths of the CIs generated by the online bootstrap method with respect to the initial step size α_0 and the learning rate η. Similarly, Figure 3 shows the sensitivity of the empirical coverage of the generated CIs with respect to the

step size parameters. These figures demonstrate that the online bootstrap method is quite robust with respect to the step size, which is one of the only user-defined parameters in the algorithm (alongside the choice of distribution for the random perturbations W_t).



(a) Sensitivity with respect to α_0 (b) Sensitivity with respect to η

Figure 2: Statistical inference in the Frozen Lake RL environment: Figure 2a shows the sensitivity of the online bootstrap CI widths with respect to the initial step size α_0. The legend specifies the values of the initial step sizes used. Similarly, Figure 2b shows the sensitivity of the CI widths with respect to the learning rate η.



(a) Sensitivity with respect to α_0 (b) Sensitivity with respect to η

Figure 3: Statistical inference in the Frozen Lake RL environment: Figure 3a shows the sensitivity of the online bootstrap CI empirical coverage with respect to the initial step size α_0. The legend specifies the values of the initial step sizes used. Similarly, Figure 3b shows the sensitivity of the empirical coverage with respect to the learning rate η.

5.2 On-policy Value Inference For Atari Pong RL Environment

In this experiment, we consider the problem of deep Q-learning (Mnih et al., 2015) in the Atari Pong environment (Brockman et al., 2016). Here the agent controls a paddle in a game

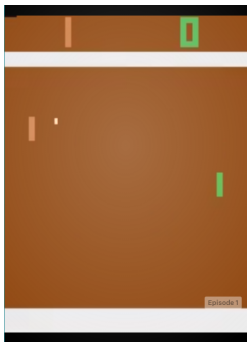
of pong, competing against a system paddle, as depicted in Figure 4a. At the end of each round, a score of 1 is awarded when a paddle hits the ball past the other paddle. The game is won by the paddle that first reaches a score of 21.

In terms of the RL environment, a reward of 1 is awarded (deducted) when the agent wins (loses) a round. An episode ends when the game is won or lost. The total reward for the episode is the net score accumulated by the agent. Rewards are scaled by 0.1 before being used as inputs to the RL agent. The state inputs to the agent are the raw pixels (RGB images) generated by the game engine. So each state is an array of shape $(210, 160, 3)$. There are 6 discrete actions available to the agent at each step.

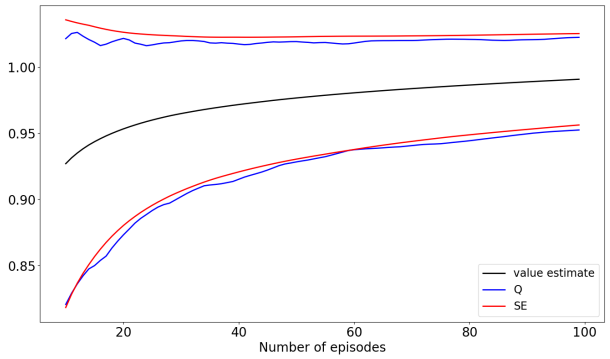
Our goal here is to learn a linear function approximation of the value function associated with a given policy. However, in this case, since the states are represented as 3-way tensors, we cannot use these states as raw features. Instead, we use a neural network to transform the states into feature vectors. We first train the agent using a Deep Q Network (DQN) with the same configuration as in Mnih et al. (2013), with three convolutional layers, and two fully connected hidden layers. The input is the pre-processed state tensor, while the outputs are the Q-function estimates for each state-action pair associated with that state.

In order to apply our algorithm to this case, we drop the output layer of the pre-trained DQN, so that the output of the resulting network is a 512-dimensional vector. These are the features we use for linear TD learning. The linear function approximation parameter θ is then a vector in $\mathbb{R}^{512}$. Aside from this feature transformation step, the application of the algorithm is the same as before. Figure 4b shows an example of the confidence intervals generated for the initial state using our algorithm over 100 episodes, depicting both quantile (Q) and standard error (SE) estimators for the confidence intervals.

Finally, Figure 5a shows the widths of the confidence intervals computed using these estimators. Figure 5b shows the error bars of the value estimates for the initial state for a number of ϵ-greedy variants of the policy learnt using deep Q-learning. This figure captures the fact that the mean value estimates decrease gradually as a function of ϵ, with the optimal



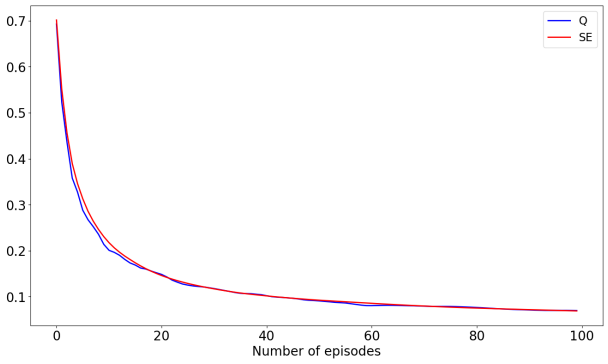
(a) Example Atari Pong game



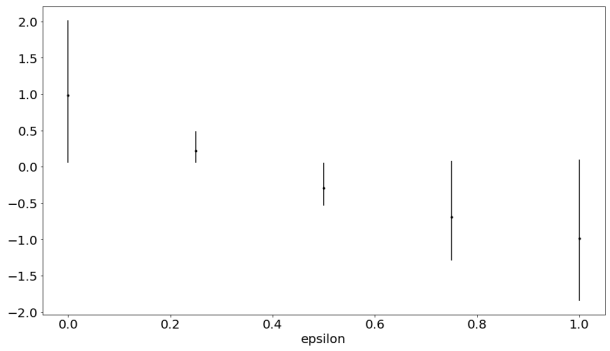
(b) Example trajectory for value estimate and confidence intervals

Figure 4: Online policy evaluation with TD learning in the Atari Pong RL environment: Figure 4a depicts an example of the Atari Pong game. This is the raw input to the learner, in the form of pixels and RGB color information. Figure 4b shows an example CI generated by the online bootstrap method for the value estimate of the initial state. Both the Quantile (Q) and Standard Error (SE) CIs are included.

policy ($\epsilon = 0$) and the random policy ($\epsilon = 1$) having the highest and lowest mean value estimates, respectively. It also shows that the policies with some amount of randomness ($\epsilon > 0$) have lower variance than the optimal policy, which reflects the fact that optimal policies are generally more prone to over-fitting, resulting in higher variance. In all these cases, the same pre-trained network was applied to compute the transformed features, which were then used as inputs to the linear TD algorithm.



(a) CI Widths



(b) 95% CI Error Bars

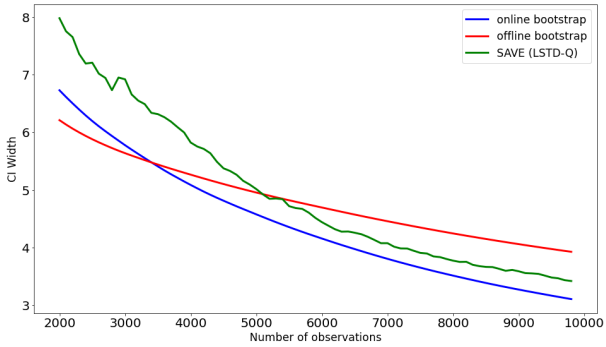
Figure 5: Online policy evaluation with TD learning in the Atari Pong RL environment: Figure 5a shows the widths of the CIs generated by the Quantile and Standard Error estimators. Figure 5b shows 95% CI error bars comparing the value estimates for 5 ϵ-greedy policies derived from the optimal policy ($\epsilon = 0$).

5.3 Off-Policy Value Inference For Infinite-Horizon MDP

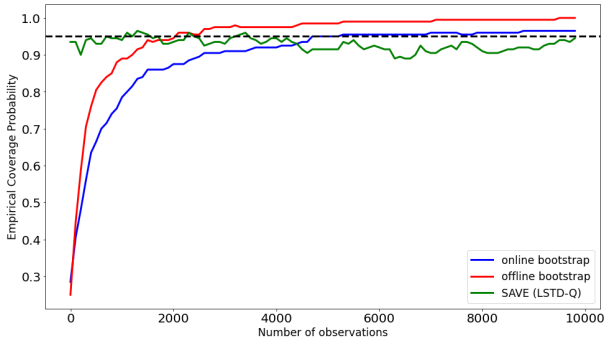
In our next experiment, we consider a simulated infinite-horizon MDP setting for off-policy evaluation. Here, we construct an environment with a state space and action space of size 50 and 5, respectively. The transition probability kernel of the MDP, the state features, and the policy we evaluate are randomly generated. We construct the true value function corresponding to the generated policy as the inner product of the state features and a randomly generated true parameter. The expected rewards at each state under the given policy are then computed using the Bellman equation (2.2). Observations are sequentially drawn according to the MDP under the generated policy, and we use this data to estimate the value of an ϵ-greedy version of the generated policy, with $\epsilon = 0.2$.

Within this framework, we implement three off-policy evaluation methods - the online bootstrap, the offline (vanilla) bootstrap, and the SAVE estimator (Shi et al., 2021). We compare the methods in terms of their CI widths and empirical coverage probabilities (computed over 200 runs) for the value estimate of one of the states. Our online bootstrap method is applied in conjunction with GTD learning (Sutton et al., 2009) to obtain the point estimates. This an online method that updates the parameters sequentially. For the offline bootstrap method, to ensure a fair comparison, we use the same GTD estimator to obtain the point estimate, and use the vanilla bootstrap to compute the confidence intervals. A similar offline bootstrap method has been used by Hanna et al. (2017) and Hao et al. (2021) for constructing their confidence intervals. The SAVE method is used here in conjunction with LSTD-Q (Lagoudakis, 2003), i.e., without the sieve basis functions. Note that LSTD-Q is a batch learning method, and therefore has different properties to GTD learning.

Figure 6 shows the results of our comparisons. Figure 6a compares the widths of the confidence intervals, while Figure 6b compares the empirical coverage probabilities over 200 simulated runs. The results show that the three methods perform roughly on par, although our online bootstrap method is much faster than the offline bootstrap, and, unlike the other two methods, does not need to store the data. Since the SAVE method uses batch LSTD-Q



(a) CI Widths



(b) Empirical Coverage Probabilities

Figure 6: Comparison of online bootstrap method to offline (vanilla) bootstrap and SAVE (LSTD-Q) methods in the infinite-horizon simulated MDP setting.

for its point estimates, it is expected to reach the desired 0.95 coverage probability in fewer observations than the other two online GTD-based methods.

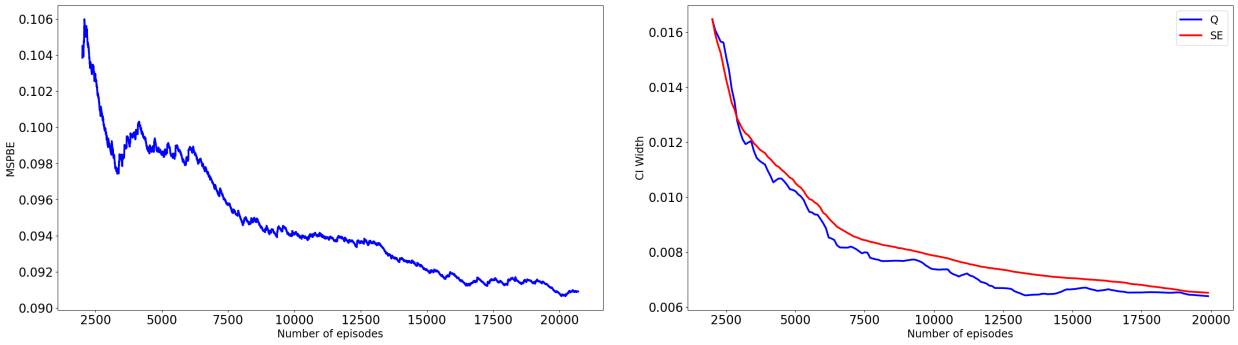
5.4 Off-Policy Value Inference For A Healthcare Application

Finally, we consider a real-world example from the healthcare domain, namely, the problem of treating sepsis in the intensive care unit. For this experiment, we use the septic management simulator by Oberst and Sontag (2019). It simulates the patient's vital signs, such as the heart rate, blood pressure, oxygen concentration, and glucose levels. There are three treatment actions (antibiotics, vasopressors, and mechanical ventilation) which the RL agent chooses from at each time step. The reward is +1 if the patient is discharged and -1 if the patient reaches a critical state. These are both absorbing states. The reward is 0 for transitions to non-absorbing states.

As in the previous experiment, we train a near-optimal policy using Q-learning. We then generate a dataset using an ϵ-greedy Q-policy, with $\epsilon = 0.05$. For the target policy, we use the optimal policy learned using Q-learning. To estimate the value function of the target policy in the off-policy setting, we use GTD learning with the Mean-Squared Projected Bellman Error (MSPBE) objective function (Sutton et al., 2009).

Since we do not have the true transition matrix, it is not possible to estimate coverage frequencies for our algorithm using the true value function. Figure 7a shows the empirical

MSPBE, which is a proxy for the estimation error for the of the point estimates. Figure 7b shows the width of the confidence intervals computed using the quantile and standard error estimators. The estimation error decreases gradually over 20,000 episodes, and the CI widths for both estimators decrease correspondingly, reflecting the decreasing uncertainty in the estimates as more data becomes available.



(a) Estimation error

(b) CI widths

Figure 7: Offline policy evaluation with GTD learning in the Septic simulation environment: Figure 7a shows the estimation error of the GTD learning algorithm, measured in terms of the Mean-Squared Projected Bellman Error (MSPBE). Figure 7b shows the width of the confidence intervals computed using the Quantile (Q) and Standard Error (SE) estimators.

6 Discussion and Future Work

In this paper, we present a fully online bootstrap algorithm for statistical inference of policy evaluation in reinforcement learning. We establish its distributional consistency in terms of the underlying algorithm, linear stochastic approximation under Markov noise. Our experimental results suggest that the online bootstrap method is efficient and effective across a range of tasks, from linear SGD with Markov noise to off-policy value estimation with GTD learning. Next, we discuss a few additional topics and interesting future directions.

6.1 Non-Stationary Behavior Policy

Our online inference method is built upon linear stochastic approximation, and focuses on two applications in RL: (1) on-policy evaluation with standard TD learning, where the

target and behavior policies are the same, and (2) off-policy evaluation with GTD learning, where the target and behavior policies are different. An interesting additional application to consider would be the case where the behavior policy changes over time, as this would enable the estimation of the value of a target policy from observations generated under a non-stationary behavior policy. To our knowledge, all existing stochastic approximation-based off-policy evaluation algorithms (GTD learning: Sutton et al. (2009), Emphatic TD: Sutton et al. (2016), and related algorithms) require the behavior policy to be stationary in order to ensure theoretically valid point estimates. That said, if these point methods were proven to be convergent under a non-stationary behavior policy, our online bootstrap-based inference method should easily be adaptable to that setting, as it only requires knowledge of the importance sampling ratio at each time step.

6.2 Semi-parametric Efficiency

The main purpose of this paper is to propose a provable online bootstrap inference method for the existing point estimators - TD learning and GTD learning, which are both special cases of linear stochastic approximation under Markov noise. In the context of i.i.d. observations, the asymptotic variance of the averaged iterate under this scheme is known to achieve the Cramer-Rao lower bound (Polyak and Juditsky, 1992; Moulines and Bach, 2011).

There are a few recent works studying the semi-parametric efficiency of reinforcement learning algorithms. For example, Ueno et al. (2011) proposed a generalized form of TD learning and studied its semi-parametric efficiency. Using this framework, they derived an optimal estimating function with the minimal asymptotic variance. In addition, a recent work by Kallus and Uehara (2021) established a lower bound for the asymptotic mean-squared error (MSE) of the Q-function estimate under a general function approximation scheme. They provide an explicit form for the approximating function that achieves the MSE lower bound. Their results require an assumption of transition sampling, i.e., state-action-reward-state tuples are drawn independently from the generative model, rather than

sequentially according to the trajectory of the MDP.

To our knowledge, there are currently no results for the efficiency of the linear stochastic approximation iterate under Markov noise. Since our focus is on the bootstrap method for uncertainty quantification, rather than the algorithm for point estimation, we leave the question of efficiency for linear stochastic approximation under Markov noise as interesting future work.

6.3 Extension to Non-Linear Function Approximation

In this work we mainly focus on the linear function approximation setting, as this is the fundamental function approximation scheme for policy evaluation algorithms in RL, serving as the basis for more elaborate approximation schemes. Here we briefly discuss two practical ways in which we can use the online bootstrap approach within the non-linear function approximation setting.

The first approach is to transform the feature basis to another space using nonlinear basis expansion, and then use bootstrapping in conjunction with linear TD learning on the transformed basis. The basis expansion can be done via sieve basis functions (Shi et al., 2021) or neural networks to learn the representation of the feature space. In our Atari Deep Q-learning experiment of Section 5.2, the raw state features are high-dimensional 3-way tensors. Here, we applied linear TD learning on the features obtained from the last layer of the pre-trained DQN to handle the non-linear function approximation. This is similar to the approach used by Chung et al. (2019), who proposed a two-timescale network architecture that enables linear methods to learn values at the top layer, with a non-linear representation learned at a slower timescale at the bottom layers.

The second approach is to directly apply our method to a non-linear function approximation scheme. The general form of the stochastic approximation is

$$\theta_{t+1} = \theta_t + \alpha_{t+1} H(\theta_t, X_{t+1}), \tag{6.1}$$

where $H(\theta, X)$ is a noisy observation of the mean field $h(\theta) = \mathbb{E}[H(\theta, X)]$. Here, the goal

is to estimate the root θ_* of the non-linear equation $h(\theta) = 0$. Under some assumptions on the non-linear function h and the iterates $\{\theta_t\}$, Andrieu et al. (2005) proved the consistency of the $\{\theta_t\}$, while Liang (2010) established a central limit theorem for the averaged iterate $\bar{\theta}_t$. In practice, we can apply our online bootstrap procedure in Algorithm 1 to produce bootstrap estimates for the iterates $\bar{\theta}_t$ under the non-linear stochastic approximation update (6.1). This allows us to perform online inference for non-linear SGD under Markov noise, which extends the online inference results for non-linear SGD under i.i.d. noise (Fang et al., 2018; Chen et al., 2020). Since the focus of this work is on linear function approximation, we leave a rigorous investigation of this non-linear setting to future work.

6.4 Approximation Error and Model Misspecification

This paper focuses on the linear function approximation of the value function. When the linear model assumption is violated, it is important to consider the model mis-specification issue in the analysis of RL algorithms with function approximation. Here we briefly discuss the implications of approximation error within the context of TD learning with linear function approximation. Our discussion starts from the least-false parameter in the linear space and then connects it with the point estimator from TD learning.

Least-false parameter in the linear space: We follow the notation defined in Section 2 of the paper. Denote $\mathcal{S}$ as the state space, Φ as the feature matrix. Let Π denote the projection operator onto the space spanned by the linear basis functions. Then, for a given policy π with a stationary distribution μ, we have

$$\Pi V^\pi = \underset{\overline{V} \in \{\Phi\theta | \theta \in \mathbb{R}^d\}}{\arg\min} \left\| V^\pi - \overline{V} \right\|_D ,$$

where D denotes the diagonal matrix with elements corresponding to the entries of the stationary distribution μ, and $\|v\| = \sqrt{v^\intercal D v}$ denotes the norm under the stationary distribution. In other words, the projected value function ΠV^π is the best approximation to the true value function V^π within the subspace spanned by the linear basis functions.

Point estimator from TD learning: Tsitsiklis and Van Roy (1997) showed that the

limiting point θ_* of the linear TD update is the unique solution to the projected Bellman equation $V_\theta = \Pi T^\pi V_\theta$, where $V_\theta = \Phi\theta$ is the value estimate corresponding to the parameter θ, while T^π denotes the Bellman operator under the policy π. In other words, the limiting point θ_* of the TD estimator can be seen as the global minimizer of the Mean-Squared Projected Bellman Error (MSPBE), i.e.,

$$\theta_* = \arg\min_{\theta \in \mathbb{R}_d} \|V_\theta - \Pi T^\pi V_\theta\|_D^2 .$$

Our online bootstrap method provides a way to perform inference on the minimizer of the MSPBE, which is a quantity of interest in its own right (Sutton and Barto, 2018).

Connection: Although the limiting point of the linear TD estimator is not the least-false parameter in the linear space, these two have a nice connection. Tsitsiklis and Van Roy (1997) showed that the value function V_{θ_*} corresponding to the limiting point θ_* satisfies

$$\|V_{\theta_*} - V^\pi\|_D \leq \frac{1}{\sqrt{1-\gamma^2}} \|\Pi V^\pi - V^\pi\|_D .$$

Thus, the approximation error for the limiting point of the TD value estimator is bounded by a constant times the approximation error for the projected value function ΠV^π, which represents the best possible approximation in the span of Φ. So, while the confidence intervals generated by the online bootstrap method provide a coverage only for the value function V_{θ_*} corresponding to the minimizer of the MSPBE, this value function itself comes with a competitive guarantee.

Acknowledgment

The authors thank the editor Professor Marina Vannucci, the associate editor and three anonymous reviewers for their valuable comments and suggestions which led to a much improved paper. Zhaoran Wang acknowledges National Science Foundation (Awards 2048075, 2008827, 2015568, 1934931), Simons Institute (Theory of Reinforcement Learning), Amazon, J.P. Morgan, and Two Sigma for their supports. Will Wei Sun's research was partially supported by ONR grant N00014-18-1-2759. Guang Cheng acknowledges support from the

National Science Foundation (NSF – SCALE MoDL (2134209)). Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of National Science Foundation and Office of Naval Research.

The authors report there are no competing interests to declare.

References

Adomavicius, G. and J. Zhang (2012). Stability of recommendation algorithms. *ACM Transactions on Information Systems (TOIS) 30*(4), 1–31.

Andrieu, C., E. Moulines, and P. Priouret (2005). Stability of stochastic approximation under verifiable conditions. *SIAM Journal on Control and Optimization 44*(1), 283–312.

Bhandari, J., D. Russo, and R. Singal (2018). A finite time analysis of temporal difference learning with linear function approximation. In S. Bubeck, V. Perchet, and P. Rigollet (Eds.), *Proceedings of the 31st Conference On Learning Theory*, Volume 75 of *Proceedings of Machine Learning Research*, pp. 1691–1692. PMLR.

Bojun, H. (2020). Steady state analysis of episodic reinforcement learning. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin (Eds.), *Advances in Neural Information Processing Systems*, Volume 33, pp. 9335–9345. Curran Associates, Inc.

Brockman, G., V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba (2016). Openai gym.

Chakraborty, B. and S. A. Murphy (2014). Dynamic treatment regimes. *Annual Review of Statistics and Its Application 1*(1), 447–464.

Chen, H., W. Lu, and R. Song (2021). Statistical inference for online decision making via stochastic gradient descent. *Journal of the American Statistical Association 116*(534), 708–719.

Chen, M., A. Beutel, P. Covington, S. Jain, F. Belletti, and E. H. Chi (2019). Top-k off-policy correction for a reinforce recommender system. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*, pp. 456–464.

Chen, X., J. D. Lee, X. T. Tong, and Y. Zhang (2020). Statistical inference for model parameters in stochastic gradient descent. *The Annals of Statistics 48*(1), 251–273.

Chen, Y., J. Fan, C. Ma, and Y. Yan (2019). Inference and uncertainty quantification for noisy matrix completion. *Proceedings of the National Academy of Sciences 116*(46), 22931–22937.

Chen, Z., S. T. Maguluri, S. Shakkottai, and K. Shanmugam (2021). A lyapunov theory for finite-sample guarantees of asynchronous q-learning and td-learning variants. *arXiv preprint arXiv:2102.01567*.

Cheng, G. (2015). Moment consistency of the exchangeably weighted bootstrap for semiparametric m-estimation. *Scandinavian Journal of Statistics 42*(3), 665–684.

Chong, E. K. P., I.-J. Wang, and S. R. Kulkarni (1999, March). Noise conditions for prespecified convergence rates of stochastic approximation algorithms. *IEEE Trans. Inf. Theory 45*(2), 810–814.

Chung, W., S. Nath, A. Joseph, and M. White (2019). Two-timescale networks for nonlinear value function approximation. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Dai, B., O. Nachum, Y. Chow, L. Li, C. Szepesvari, and D. Schuurmans (2020). Coindice: Off-policy confidence interval estimation. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin (Eds.), *Advances in Neural Information Processing Systems*, Volume 33, pp. 9398–9411. Curran Associates, Inc.

DasGupta, A. (2008). *The Bootstrap*, pp. 461–497. New York, NY: Springer New York.

Delyon, B. (2000). Stochastic approximation with decreasing gain: Convergence and asymptotic theory.

Douc, R., E. Moulines, P. Priouret, and P. Soulier (2018). *Markov chains*. Operation research and financial engineering. Springer.

Dulac-Arnold, G., N. Levine, D. J. Mankowitz, J. Li, C. Paduraru, S. Gowal, and T. Hester (2021, Sep). Challenges of real-world reinforcement learning: definitions, benchmarks and analysis. *Machine Learning 110*(9), 2419–2468.

Durmus, A., E. Moulines, A. Naumov, S. Samsonov, and H.-T. Wai (2021). On the stability of random matrix product with markovian noise: Application to linear stochastic approximation and td learning. In *Conference on Learning Theory*, pp. 1711–1752. PMLR.

Ertefaie, A. and R. L. Strawderman (2018). Constructing dynamic treatment regimes over indefinite time horizons. *Biometrika 105*(4), 963–977.

Fang, Y., J. Xu, and L. Yang (2018). Online bootstrap confidence intervals for the stochastic gradient descent estimator. *Journal of Machine Learning Research 19*(78), 1–21.

Gu, S., E. Holly, T. Lillicrap, and S. Levine (2017). Deep reinforcement learning for robotic manipulation with asynchronous off-policy updates. In *2017 IEEE international conference on robotics and automation (ICRA)*, pp. 3389–3396. IEEE.

Gupta, H., R. Srikant, and L. Ying (2019). Finite-time performance bounds and adaptive learning rate selection for two time-scale reinforcement learning. In H. M. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. B. Fox, and R. Garnett (Eds.), *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pp. 4706–4715.

Hall, P. (1992). *The Bootstrap and Edgeworth Expansion*. Springer New York.

Hanna, J. P., P. Stone, and S. Niekum (2017). Bootstrapping with models: Confidence intervals for off-policy evaluation. In *Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems*, AAMAS '17, Richland, SC, pp. 538–546. International Foundation for Autonomous Agents and Multiagent Systems.

Hao, B., Y. Abbasi Yadkori, Z. Wen, and G. Cheng (2019). Bootstrapping upper confidence bound. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (Eds.), *Advances in Neural Information Processing Systems*, Volume 32. Curran Associates, Inc.

Hao, B., X. Ji, Y. Duan, H. Lu, C. Szepesvári, and M. Wang (2021). Bootstrapping statistical inference for off-policy evaluation. *arXiv preprint arXiv:2102.03607*.

Hu, B. and U. A. Syed (2019). Characterizing the exact behaviors of temporal difference learning algorithms using markov jump linear system theory. In H. M. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. B. Fox, and R. Garnett (Eds.), *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pp. 8477–8488.

Jiang, N. and J. Huang (2020). Minimax value interval for off-policy evaluation and policy optimization. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin (Eds.), *Advances in Neural Information Processing Systems*, Volume 33, pp. 2747–2758. Curran Associates, Inc.

Kaledin, M., E. Moulines, A. Naumov, V. Tadic, and H.-T. Wai (2020). Finite time analysis of linear two-timescale stochastic approximation with markovian noise. In *Conference on Learning Theory*, pp. 2144–2203. PMLR.

Kallus, N. and M. Uehara (2021). Efficiently breaking the curse of horizon in off-policy evaluation with double reinforcement learning.

Kuzborskij, I., C. Vernade, A. Gyorgy, and C. Szepesvari (2021). Confident off-policy evaluation and selection through self-normalized importance weighting. In A. Banerjee and K. Fukumizu (Eds.), *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*, Volume 130 of *Proceedings of Machine Learning Research*, pp. 640–648. PMLR.

Lagoudakis, M. G. (2003). Least-squares policy iteration. *Journal of Machine Learning Research 4*, 1107–1149.

Levin, D., Y. Peres, and E. L. Wilmer (2017). *Markov Chains and Mixing Times : Second Edition*. Providence: American Mathematical Society.

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

Li, T., L. Liu, A. Kyrillidis, and C. Caramanis (2018). Statistical inference using sgd. *Proceedings of the AAAI Conference on Artificial Intelligence 32*(1).

Li, Y., H. Xie, Y. Lin, and J. C. Lui (2021). Unifying offline causal inference and online bandit learning for data driven decision. In *Proceedings of the Web Conference 2021*, pp. 2291–2303.

Liang, F. (2010). Trajectory averaging for stochastic approximation mcmc algorithms. *The Annals of Statistics 38*(5), 2823–2856.

Luckett, D. J., E. B. Laber, A. R. Kahkoska, D. M. Maahs, E. Mayer-Davis, and M. R. Kosorok (2019). Estimating dynamic treatment regimes in mobile health using v-learning. *Journal of the American Statistical Association*.

Meyn, S. and R. L. Tweedie (2009). *Markov Chains and Stochastic Stability* (2nd ed.). USA: Cambridge University Press.

Mnih, V., K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller (2013). Playing atari with deep reinforcement learning. cite arxiv:1312.5602Comment: NIPS Deep Learning Workshop 2013.

Mnih, V., K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, et al. (2015). Human-level control through deep reinforcement learning. *nature 518*(7540), 529–533.

Mou, W., A. Pananjady, M. J. Wainwright, and P. L. Bartlett (2021). Optimal and instance-dependent guarantees for markovian linear stochastic approximation. *arXiv preprint arXiv:2112.12770*.

Moulines, E. and F. Bach (2011). Non-asymptotic analysis of stochastic approximation algorithms for machine learning. In J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, and K. Q. Weinberger (Eds.), *Advances in Neural Information Processing Systems*, Volume 24. Curran Associates, Inc.

Oberst, M. and D. Sontag (2019). Counterfactual off-policy evaluation with Gumbel-max structural causal models. In K. Chaudhuri and R. Salakhutdinov (Eds.), *Proceedings of the 36th International Conference on Machine Learning*, Volume 97 of *Proceedings of Machine Learning Research*, pp. 4881–4890. PMLR.

Parekh, V. S. and M. A. Jacobs (2019). Deep learning and radiomics in precision medicine. *Expert review of precision medicine and drug development 4*(2), 59–72.

Polyak, B. T. and A. B. Juditsky (1992). Acceleration of stochastic approximation by averaging. *SIAM Journal on Control and Optimization 30*(4), 838–855.

Robbins, H. and S. Monro (1951). A Stochastic Approximation Method. *The Annals of Mathematical Statistics 22*(3), 400 – 407.

Ruppert, D. (1988). Efficient estimations from a slowly convergent robbins-monro process. Technical report, Cornell University Operations Research and Industrial Engineering.

Sallab, A. E., M. Abdou, E. Perot, and S. Yogamani (2017). Deep reinforcement learning framework for autonomous driving. *Electronic Imaging 2017*(19), 70–76.

Shi, C., S. Luo, H. Zhu, and R. Song (2021). An online sequential test for qualitative treatment effects. *Journal of Machine Learning Research 22*(286), 1–51.

Shi, C., X. Wang, S. Luo, H. Zhu, J. Ye, and R. Song (2022). Dynamic causal effects evaluation in a/b testing with a reinforcement learning framework. *Journal of the American Statistical Association* (just-accepted), 1–29.

Shi, C., S. Zhang, W. Lu, and R. Song (2021). Statistical inference of the value function for reinforcement learning in infinite horizon settings. *Journal of the Royal Statistical Society, Series B*.

Srikant, R. and L. Ying (2019). Finite-time error bounds for linear stochastic approximation andtd learning. In *Conference on Learning Theory*, pp. 2803–2830. PMLR.

Sutton, R. S. (1988). Learning to predict by the methods of temporal differences. *Machine learning 3*(1), 9–44.

Sutton, R. S. and A. G. Barto (2018). *Reinforcement learning: An introduction*. MIT press.

Sutton, R. S., H. R. Maei, D. Precup, S. Bhatnagar, D. Silver, C. Szepesvári, and E. Wiewiora (2009). Fast gradient-descent methods for temporal-difference learning with linear function approximation. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pp. 993–1000.

Sutton, R. S., H. R. Maei, D. Precup, S. Bhatnagar, D. Silver, C. Szepesvári, E. Wiewiora, and M. R. Introduced (2009). Fast gradient-descent methods for temporal-difference learning with linear function approximation. In *In Danyluk et*, pp. 993–1000.

Sutton, R. S., A. R. Mahmood, and M. White (2016). An emphatic approach to the problem of off-policy temporal-difference learning. *Journal of Machine Learning Research 17*(73), 1–29.

Sutton, R. S., C. Szepesvári, and H. R. Maei (2008). A convergent o(n) algorithm for off-policy temporal-difference learning with linear function approximation. In *Proceedings of the 21st International Conference on Neural Information Processing Systems*, NIPS'08, USA, pp. 1609–1616. Curran Associates Inc.

Tsitsiklis, J. N. and B. Van Roy (1997, May). An analysis of temporal-difference learning with function approximation. *IEEE Trans. Automat. Contr. 42*(5), 674–690.

Ueno, T., S.-i. Maeda, M. Kawanabe, and S. Ishii (2011). Generalized td learning. *Journal of Machine Learning Research 12*(6).

van der Vaart, A. W. (1998). *Asymptotic Statistics*. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press.

Wang, C.-H., Y. Yu, B. Hao, and G. Cheng (2020). Residual bootstrap exploration for bandit algorithms. *arXiv preprint arXiv:2002.08436*.

White, M. and A. White (2010). Interval estimation for reinforcement-learning algorithms in continuous-state domains. In J. Lafferty, C. Williams, J. Shawe-Taylor, R. Zemel, and A. Culotta (Eds.), *Advances in Neural Information Processing Systems*, Volume 23. Curran Associates, Inc.

Xu, T., Z. Wang, Y. Zhou, and Y. Liang (2020). Reanalysis of variance reduced temporal difference learning. In *International Conference on Learning Representations*.

Xu, Z., Z. Li, Q. Guan, D. Zhang, Q. Li, J. Nan, C. Liu, W. Bian, and J. Ye (2018). Large-scale order dispatch in on-demand ride-hailing platforms: A learning and planning approach. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 905–913.

Zhang, K. W., L. Janson, and S. A. Murphy (2021). Statistical inference with m-estimators on bandit data. *arXiv preprint arXiv:2104.14074*.

SUPPLEMENTARY MATERIAL

Online Bootstrap Inference For Policy Evaluation in Reinforcement Learning

In this online supplementary material, we provide detailed proofs for the lemmas and main theorems in Sections 4.1 and 4.2, as well as additional experiments.

S1 Proofs for Section 4.1

The following lemma is a restatement of Theorem 2 from Chong et al. (1999). In the following, we say that a sequence ϵ_t is *small* with respect to another sequence α_t if there exist sequences $\{e_t\}$ and $\{r_t\}$ such that $\epsilon_t = e_t + r_t$ for all t, $r_t \to 0$, and $\sum_{k=1}^{t} \alpha_t \|e_t\|_2$ converges. Also, we say that a scalar sequence $\{a_t\}$ has *bounded variation* if $\sum_{t=1}^{\infty} |a_{t+1} - a_t| < \infty$. We refer to the cited article for further details on these conditions.

Lemma S1.1. *Consider the linear stochastic approximation update*

$$\theta_{t+1} = \theta_t + \alpha_{t+1}(\tilde{A}(X_{t+1})\theta_t - \tilde{b}(X_{t+1})).$$

Assume the following conditions hold:

(B1) The step size sequence $\{\alpha_t\}$ *satisfies* $\alpha_t > 0$, $\alpha_t \to 0$, *and* $\sum_{t=1}^{\infty} \alpha_t = \infty$.

(B2) $\bar{A}$ *is a bounded Hurwitz matrix.*

(B3) $\{\tilde{A}(X_t) - \bar{A}\}$ *is small with respect to* α_t.

(B4) Let $\{\rho_t\}$ *be a positive real sequence converging monotonically to 0, such that*

(i) $\{\rho_t^{-1}(\tilde{b}(X_t) - \bar{b})\}$ *is small with respect to* α_t.

(ii) $(\rho_t - \rho_{t+1})/(\alpha_t \rho_t) \to c < \infty$,

(iii) The sequences $\{\rho_{t+1}/\rho_t\}$ *and* $\{\rho_t/\rho_{t+1}\}$ *have bounded variation.*

Then $\theta_t - \theta_* = o(\rho_t)$.

Proof of Proposition 4.1

We verify that the conditions of Lemma S1.1 hold under our assumptions (A1), (A2), (A3).

Firstly, it is easy to see that (B1) holds under (A3), i.e., with a step size $\alpha_t = \alpha_0 t^{-\eta}$, $\eta \in (1/2, 1)$. Similarly, (B2) follows directly from assumption (A2).

For functions f defined over the state space $\mathcal{X}$, we define the t-step transition operator $\mathcal{P}^t f(x) = \int_{y\in\mathcal{X}} f(y)\mathcal{P}^t(x, dy)$, where $\mathcal{P}^t(x, y)$ denotes the t-step transition probability from state x to y. When $t = 1$, we write $\mathcal{P}^1 f(x) = \mathcal{P}f(x)$.

Next, define $\hat{A} : \mathcal{X} \to \mathbb{R}^{d\times d}$ and $\hat{b} : \mathcal{X} \to \mathbb{R}^d$ to be the solutions to the Poisson equations

$$\tilde{A}(x) - \bar{A}(x) = \hat{A}(x) - \mathcal{P}\hat{A}(x),$$
$$\tilde{b}(x) - \bar{b}(x) = \hat{b}(x) - \mathcal{P}\hat{b}(x),$$

for $x \in \mathcal{X}$. The existence of $\hat{A}$ and $\hat{b}$ is guaranteed under (A1). Furthermore, under (A2), there exist constants $\hat{A}_{\max}, \hat{b}_{\max>0}$ such that $\|\hat{A}\|_F \le \hat{A}_{\max}$ and $\|\hat{b}\|_2 \le \hat{b}_{\max}$ (Delyon, 2000).

Then we can write $\tilde{A}(X_t) - \bar{A} = e_t + r_t$, where $e_t = \hat{A}(X_t) - \mathcal{P}\hat{A}(X_{t+1})$, and $r_t = \mathcal{P}\hat{A}(X_{t+1}) - \mathcal{P}\hat{A}(X_t)$. To verify condition (B3), it suffices to show that $\sum_{t=1}^{\infty} \alpha_t\|e_t\| < \infty$, and $\|r_t\| \to 0$, as $t \to \infty$ (Chong et al., 1999).

Let $\mathcal{F}_t = \sigma(\{X_t\})$ denote the natural filtration with respect to the Markov chain X_t. Then $\mathbb{E}[e_t|\mathcal{F}_t] = 0$, and e_t is a martingale difference sequence with respect to $\mathcal{F}_t$. Furthermore, e_t is a.s. uniformly bounded, since $\|e_t\|_F \le 2\hat{A}_{\max}$, by construction. So $\sum_{t=1}^{\infty} \alpha_t^2\mathbb{E}[\|e_t\|^2|\mathcal{F}_t] < \infty$. Then, by Theorem 29 of Delyon (2000), $\sum_{t=1}^{\infty} \alpha_t\|e_t\|$ converges.

Also, since $\mathcal{P}(X_t, \cdot) \to \mu$ as $t \to \infty$, it follows that $\|r_t\| \to 0$, as $t \to \infty$. So $\{\tilde{A}(X_t) - \bar{A}\}$ is small with respect to α_t, and (B3) holds.

Next, set $\rho_t = t^{-\gamma}$, with $\gamma \in (0, \eta - 1/2)$. Conditions (B4)(ii) and (B4)(iii) hold under this definition, with $c = 0$ in (B4)(ii) (Chong et al., 1999). It remains to verify (B4)(i).

Define $\tilde{b}(X_t) - \bar{b} = e_t + r_t$, where $e_t = \hat{b}(X_t) - \mathcal{P}\hat{b}(X_{t+1})$, and $r_t = \mathcal{P}\hat{b}(X_{t+1}) - \mathcal{P}\hat{b}(X_t)$. It suffices to show that $\sum_{t=1}^{\infty} \alpha_t\rho_t^{-1}\|e_t\| < \infty$, and that $\rho_t^{-1}\|r_t\| \to 0$, as $t \to \infty$.

By the same argument used above for $\{\tilde{A} - \bar{A}\}$, e_t is an a.s. uniformly bounded martingale

difference sequence, with $\|e_t\|_F \leq 2\hat{b}_{\max}$ for all t. Then $\sum_{t=1}^{\infty} \alpha_t^2 \rho_t^{-2} \mathbb{E}[\|e_t\|_2 | \mathcal{F}_t] < \infty$, since $\eta - \gamma > 1/2$. So, by Theorem 29 of Delyon (2000), $\sum_{t=1}^{\infty} \alpha_t \rho_t \|e_t\|$ converges.

Next, we have

$$\begin{aligned}
\|\mathcal{P}\hat{b}(X_{t+1}) - \mathcal{P}\hat{b}(X_t)\| &= \left\| \int_{y \in \mathcal{X}} \hat{b}(y)(\mathcal{P}(X_{t+1}, dy) - \mathcal{P}(X_t, dy)) \right\| \\
&\leq \int_{y \in \mathcal{X}} \|\hat{b}(y)\| \|\mathcal{P}(X_{t+1}, dy) - \mathcal{P}(X_t, dy)\| \\
&\leq \hat{b}_{\max} \int_{y \in \mathcal{X}} \|\mathcal{P}(X_{t+1}, dy) - \mathcal{P}(X_t, dy)\|.
\end{aligned}$$

Consider the integrand in the above expression. For any bounded initial distribution ν_0, i.e., with $\sup_{x \in \mathcal{X}} \|\nu_0(x)\| \leq \nu_{\max}$, for some constant $\nu_{\max} < \infty$, we have

$$\begin{aligned}
\|\mathcal{P}(X_{t+1}, dy) - \mathcal{P}(X_t, dy)\| &\leq \sup_{x_1, x_2 \in \mathcal{X}} \|\nu_0 \mathcal{P}^{t+1}(x_1, dy) - \nu_0 \mathcal{P}^t(x_2, dy)\| \\
&\leq \nu_{\max} \sup_{x_1, x_2 \in \mathcal{X}} \|\mathcal{P}^{t+1}(x_1, dy) - \mathcal{P}^t(x_2, dy)\| \\
&= \nu_{\max} \sup_{x_1, x_2 \in \mathcal{X}} \|(\mathcal{P}^{t+1}(x_1, dy) - \pi(dy)) - (\mathcal{P}^t(x_2, dy) - \pi(dy))\| \\
&\leq \nu_{\max} \sup_{x_1, x_2 \in \mathcal{X}} \left(\|\mathcal{P}^{t+1}(x_1, dy) - \pi(dy)\| + \|\mathcal{P}^t(x_2, dy) - \pi(dy)\| \right) \\
&\leq \nu_{\max} \left(M\kappa^{t+1} + M\kappa^t \right) \\
&\leq 2\nu_{\max} M \kappa^t,
\end{aligned}$$

where the penultimate inequality holds by (4.1). It follows that $\|\mathcal{P}\hat{b}(X_{t+1}) - \mathcal{P}\hat{b}(X_t)\| \leq 2\hat{b}_{\max}\nu_{\max} M\kappa^t$, and so $\rho_t^{-1}\|r_t\| \leq 2\hat{b}_{\max}\nu_{\max} M t^{\gamma} \kappa^t \to 0$, as $t \to \infty$. □

Proof of Proposition 4.2

First, we list the conditions required for our central limit theorem, Proposition 4.2, to hold. The assumptions listed below are from Liang (2010), who proved a central limit theorem for the varying truncation stochastic approximation MCMC algorithm. This is a general form of algorithm (2.1), and is designed to solve the equation

$$h(\theta) = \int_{\mathcal{X}} H(\theta, x) f_\theta(x) dx = 0,$$

where $\theta \in \Theta \subset \mathbb{R}^{d_\theta}$ is a parameter vector and $f_\theta(x), x \in \mathcal{X} \subset \mathbb{R}^{d_x}$ is a density function depending on θ. The function $h(\theta)$ is called the mean field function, and $H(\theta, x)$ is a noisy observation of $h(\theta)$.

The stochastic approximation algorithm is designed to iteratively estimate θ from a sequence of noisy observations that depend on the current estimate of θ (hence forming a controlled Markov chain). The main update step for this algorithm is given by

$$\begin{aligned} \theta_{t+1} &= \theta_t + \alpha_{t+1} H(\theta_t, X_{t+1}) \\ &= \theta_t + \alpha_{t+1} h(\theta_t) + \alpha_{t+1} \epsilon_{t+1}, \end{aligned} \tag{S1.1}$$

where $h(\theta) = \int H(\theta, x) f_\theta(x) dx$, f_θ being the invariant distribution of the controlled Markov transition kernel $\mathcal{P}_\theta$, and $\epsilon_{t+1} = H(\theta_t, X_{t+1}) - h(\theta_t)$ is the residual noise term.

In order to ensure the convergence of the iterates in (S1.1), Liang (2010) imposes a varying truncation scheme, whereby the iterates θ_t are constrained within an increasing sequence of compact sets $\{\mathcal{K}_s\}_{s \geq 0}$. Under this scheme, Andrieu et al. (2005) showed that there exists a time step $t_{\sigma_s} < \infty$ such that $\theta_t \in \mathcal{K}_{\sigma_s}$ for all $t \geq t_{\sigma_s}$, and there are no further truncations beyond time step t_{σ_s}. The central limit theorem applies to the averaged iterate $\bar{\theta}_t := \frac{1}{t - t_{\sigma_s}} \sum_{i=t_{\sigma_s}+1}^{t} \theta_i$.

The following conditions are assumed by Liang (2010):

(C1) Θ is an open set, the function $h : \Theta \to \mathbb{R}^d$ is continuous, and there exists a continuously differential function $v : \Theta \to [0, \infty)$ such that

(i) There exists $M_0 > 0$ such that

$$\mathcal{L} = \{\theta \in \Theta, \langle \nabla v(\theta), h(\theta) \rangle = 0\} \subset \{\theta \in \Theta, v(\theta) < M_0\}.$$

(ii) There exists $M_1 \in (M_0, \infty)$ such that $\mathcal{V}_{M_1}$ is a compact set, where $\mathcal{V}_M = \{\theta \in \Theta, v(\theta) \leq M\}$.

(iii) For any $\theta \in \Theta \backslash \mathcal{L}$, $\langle \nabla v(\theta), h(\theta) \rangle < 0$.

(iv) The closure of $v(\mathcal{L})$ has an empty interior.

(C2) The mean field $h(\theta)$ is measurable and locally bounded. There exists a Hurwitz matrix F, $\gamma > 0, \rho \in (0, 1]$, and a constant c such that, for any $\theta_* \in \mathcal{L}$,

$$\|h(\theta) - F(\theta - \theta_*)\| \leq c\|\theta - \theta^*\|^{1+\rho} \quad \forall \theta \in \{\theta : \|\theta - \theta_*\| \leq \gamma\},$$

where $\mathcal{L}$ is defined in (B1)(i).

(C3) For any $\theta \in \Theta$, the transition kernel $\mathcal{P}_\theta$ is irreducible and aperiodic. In addition, there exists a function $V : \mathcal{X} \to [1, \infty)$, and a constant $\alpha \geq 2$ such that for any compact set $\mathcal{K} \subset \Theta$:

(i) There exists a set $\mathbf{C} \subset \mathcal{X}$, and integer l, constants $0 < \lambda < 1, b, \zeta, \delta > 0$ and a probability measure ν such that

$$\begin{aligned}
&\sup_{\theta \in \mathcal{K}} \mathcal{P}_\theta^l V^\alpha(x) \leq \lambda V^\alpha(x) + bI(x \in \mathbf{C}) \quad \forall x \in \mathcal{X}, \\
&\sup_{\theta \in \mathcal{K}} \mathcal{P}_\theta V^\alpha(x) \leq \zeta V^\alpha(x) \quad \forall x \in \mathcal{X}, \\
&\inf_{\theta \in \mathcal{K}} \mathcal{P}_\theta^l(x, A) \geq \delta \nu(A) \quad \forall x \in \mathbf{C}, \forall A \in \mathcal{B}_\mathcal{X}.
\end{aligned}$$

(ii) There exists a constant $c > 0$ such that, for all $x \in \mathcal{X}$,

$$\begin{aligned}
&\sup_{\theta \in \mathcal{K}} \|H(\theta, x)\|_V \leq c, \\
&\sup_{\theta, \theta' \in \mathcal{K}} \|H(\theta, x) - H(\theta', x)\|_V \leq c\|\theta - \theta'\|.
\end{aligned}$$

(iii) There exists a constant $c > 0$ such that, for all $\theta, \theta' \in \mathcal{K}$,

$$\begin{aligned}
&\|\mathcal{P}_\theta g - \mathcal{P}_{\theta'}\|_V \leq c\|g\|_V\|\theta - \theta'\| \quad \forall g \in \mathcal{L}_V, \\
&\|\mathcal{P}_\theta g - \mathcal{P}_{\theta'} g\|_{V^\alpha} \leq c\|g\|_{V^\alpha}\|\theta - \theta'\| \quad \forall g \in \mathcal{L}_{V^\alpha}.
\end{aligned}$$

(C4) The step sizes $\{\alpha_t\}$ are non-increasing, positive sequences that satisfy the conditions

$$\sum_{t=1}^{\infty} \alpha_t = \infty, \quad \lim_{t \to \infty} (t\alpha_t) = \infty, \quad \frac{\alpha_{t+1} - \alpha_t}{\alpha_t} = o(\alpha_{t+1}), \quad \sum_{t=1}^{\infty} \frac{\alpha_t^{(1+\tau)/2}}{\sqrt{t}} < \infty,$$

for some $\tau \in (0, 1]$ and a constant $\alpha \geq 2$ defined in (B3).

We refer to Liang (2010) for further details on these conditions.

We now verify that (C1)-(C4) hold under assumptions (A1)-(A3). We also show that, under our assumptions, the iterates of the update (2.1) are constrained within a compact set $\mathcal{K} \subset \Theta$, thereby avoiding the need for the varying truncation scheme. Then the result directly follows. Using the notation of (S1.1), in our case, we have $H(\theta, x) = \tilde{A}(x)\theta - \tilde{b}$, with mean field $h(\theta) = \bar{A}\theta - \bar{b}$. By (A2)(iii), we have $\mathbb{E}_{X\sim\mu}[H(\theta, X)] = h(\theta)$, for all $\theta \in \Theta$.

(C1) assumes the existence of a global Lyapunov function v. We may choose $v(\theta) = \theta^\intercal \bar{b} - \frac{1}{2}\theta^\intercal \bar{A}\theta$. Then v is a global Lyupanov function for the mean field h (Andrieu et al., 2005; Liang, 2010). $\mathcal{L}$ denotes the set of all valid solutions θ^* for the equation $h(\theta) = 0$. In our case, since $\bar{A}$ is Hurwitz by (A2)(ii), there exists a unique solution θ^* for the linear system $\bar{A}\theta = \bar{b}$, and $\mathcal{L}$ is a singleton set.

For (C2), the measurability and local boundedness of h follows directly from linearity. For the latter part, we may choose $F = A$. Then for any $\theta \in \Theta$, we have $\|h(\theta) - F(\theta - \theta^*)\| \equiv 0$, so (C2) holds.

For (C3), in our case the function $H(\theta, x) = \tilde{A}(x)\theta - \tilde{b}(x)$ is bounded by (A2)(ii), so we can choose the drift function $V \equiv 1$. Then the first two conditions of (C3)(i) hold trivially.

The third condition in (C3)(i) is a standard assumption in the Markov Chain Monte Carlo (MCMC) literature, and is referred to as the minorization condition. By Theorem 5.2.2 of Meyn and Tweedie (2009), for φ-irreducible Markov chains, "small sets" for which the minorization condition holds exist. By (A1), the Markov chain is irreducible, and so, by definition, is φ-irreducible for some irreducibility measure φ. Hence the condition holds in our case.

(C3)(ii) follows directly from (A2)(ii). (C3)(iii) does not apply in our case as we are dealing with a homogeneous Markov chain that does not depend on θ_k (not have a controlled Markov chain). The conditions of (C4) hold trivially under (A3).

Finally, by Proposition 4.1, we can choose a large enough constant $R_\theta > 0$ such that $\|\theta_t - \theta_*\|_2 \leq R_\theta$ for all $t \geq 0$. Then $\theta_t \in \mathcal{K}$ for all $t \geq 0$, for some compact set $\mathcal{K} \subset \Theta$. □

S2 Proofs for Section 4.2

Proof of Proposition 4.3

To verify the conditions for Lemma S1.1, it suffices to check that $\{W_t\tilde{A}(X_t) - \bar{A}\}$ and $\{\rho_t^{-1}(W_t\tilde{b}(X_t) - \bar{b})\}$ are small with respect to the step sizes $\{\alpha_t\}$. By (A2)(ii) and the boundedness of W_t, we have $\|W_t\tilde{A}(X_t)\|_F \le W_{\max}A_{\max} < \infty$, and $\|W_t\tilde{b}(X_t)\| \le W_{\max}b_{\max} < \infty$, for all $t \ge 1$. By independence of W_t, we have $\mathbb{E}_\mu[W_t\tilde{A}(X_t) - \bar{A}] = 0$ and $\mathbb{E}_\mu[W_t\tilde{b}(X_t) - \bar{b}] = 0$. The rest of the argument is identical to the proof of Proposition 4.1. □

Lemma S2.1. *Assume (A1)-(A3) hold. Then*

$$\sqrt{t}(\bar{\hat{\theta}}_t - \theta_*) = -\bar{A}^{-1}\frac{1}{\sqrt{t}}\sum_{i=1}^{t}\hat{\epsilon}_{i+1} + o_p(1).$$

Proof. An argument similar to above may be used to verify that conditions (C1)-(C4) also hold for the perturbed SA update (3.1) under assumptions (A1)-(A3). Then the result follows as an intermediate step in the proof of Theorem 2.2 by Liang (2010). □

The following lemma is adapted from Lemma 5 of Xu et al. (2020).

Lemma S2.2. *Assume (A1)-(A3) hold. Then, for any $i > j$, we have*

$$\left\|\mathbb{E}\left[\tilde{A}(X_i)|\mathcal{F}_j\right] - \bar{A}\right\|_F \le A_{\max}M\kappa^{i-j},$$

where M and κ refer to the constants from (4.1).

Proof. By (4.1), for any $i > j$, the following holds:

$$\left\|\mathcal{P}^{i-j}(\cdot|\mathcal{F}_j) - \mu\right\| \le M\kappa^{i-j}, \tag{S2.1}$$

Then we have

$$\begin{aligned}
\left\|\mathbb{E}\left[\tilde{A}(X_i)|\mathcal{F}_j\right] - \bar{A}\right\|_F &= \left\|\int_{x\in\mathcal{X}}\tilde{A}(x)\mathcal{P}^{i-j}(dx|\mathcal{F}_j) - \int_{x\in\mathcal{X}}\tilde{A}(x)\mu(dx)\right\|_F \\
&\le \int_{x\in\mathcal{X}}\left\|\tilde{A}(x)\mathcal{P}^{i-j}(dx|\mathcal{F}_j) - \tilde{A}(x)\mu(dx)\right\|_F \\
&\le \int_{x\in\mathcal{X}}\left\|\tilde{A}(x)\right\|_F\left\|\mathcal{P}^{i-j}(dx|\mathcal{F}_j) - \mu(dx)\right\| \\
&\le A_{\max}M\kappa^{i-j}.
\end{aligned}$$

The first equality follows from the definition of $\bar{A}$ in (A2)(i), the second step holds by Jensen's inequality, and the final step follows from (A2)(iii) and (S2.1). □

Proof of Lemma 4.1

Starting with (4.4), we have

$$\begin{aligned}\hat{\epsilon}_{t+1} &= (W_{t+1}\tilde{A}(X_{t+1}) - \bar{A})\hat{\theta}_t - (W_{t+1}\tilde{b}(X_{t+1}) - \bar{b}) \\ &= W_{t+1}(\tilde{A}(X_{t+1})\theta_* - \tilde{b}(X_{t+1})) + (W_{t+1}\tilde{A}(X_{t+1}) - \bar{A})(\hat{\theta}_t - \theta_*). \qquad \text{(S2.2)}\end{aligned}$$

using the fact that $\bar{A}\theta_* = \bar{b}$.

By Lemma S2.1 and (S2.2), we have

$$\begin{aligned}&\sqrt{t}(\bar{\hat{\theta}}_t - \theta_*) = -\bar{A}^{-1}\frac{1}{\sqrt{t}}\sum_{i=1}^{t}\hat{\epsilon}_{i+1} + o_p(1) \\ &= -\bar{A}^{-1}\frac{1}{\sqrt{t}}\sum_{i=1}^{t}W_{i+1}(\tilde{A}(X_{i+1})\theta_* - \tilde{b}(X_{i+1})) - \bar{A}^{-1}\frac{1}{\sqrt{t}}\sum_{i=1}^{t}(W_{i+1}\tilde{A}(X_{i+1}) - \bar{A})(\hat{\theta}_i - \theta_*) + o_p(1).\end{aligned}$$

Consider the second term in the above expression. We want to show that this term is $o_p(1)$. It suffices to show that its second moment vanishes as $t \to \infty$. First we expand the second moment and split it into square and cross terms. We have

$$\begin{aligned}&\mathbb{E}\left[\left\|\frac{1}{\sqrt{t}}\sum_{i=1}^{t}\left(W_{i+1}\tilde{A}(X_{i+1}) - \bar{A}\right)\left(\hat{\theta}_i - \theta_*\right)\right\|_2^2\right] \\ &= \frac{1}{t}\sum_{i=1}^{t}\sum_{j=1}^{t}\mathbb{E}\left[\left\langle\left(W_{i+1}\tilde{A}(X_{i+1}) - \bar{A}\right)\left(\hat{\theta}_i - \theta_*\right), \left(W_{j+1}\tilde{A}(X_{j+1}) - \bar{A}\right)\left(\hat{\theta}_j - \theta_*\right)\right\rangle\right] \\ &= \frac{1}{t}\sum_{i=1}^{t}\mathbb{E}\left[\left\|\left(W_{i+1}\tilde{A}(X_{i+1}) - \bar{A}\right)\left(\hat{\theta}_i - \theta_*\right)\right\|_2^2\right] \\ &+ \frac{1}{t}\sum_{i\neq j}\mathbb{E}\left[\left\langle\left(W_{i+1}\tilde{A}(X_{i+1}) - \bar{A}\right)\left(\hat{\theta}_i - \theta_*\right), \left(W_{j+1}\tilde{A}(X_{j+1}) - \bar{A}\right)\left(\hat{\theta}_j - \theta_*\right)\right\rangle\right] \\ &= I_1 + I_2.\end{aligned}$$

We deal with each term separately. First, we have

$$I_1 = \frac{1}{t}\sum_{i=1}^{t}\mathbb{E}\left[(\hat{\theta}_i - \theta_*)^\intercal (W_{i+1}\tilde{A}(X_{i+1}) - \bar{A})^\intercal (W_{i+1}\tilde{A}(X_{i+1}) - \bar{A})(\hat{\theta}_i - \theta_*)\right]$$
$$\leq \frac{\lambda_A}{t}\sum_{i=1}^{t}\mathbb{E}\left[\left\|\hat{\theta}_i - \theta_*\right\|_2^2\right] \to 0,$$

since $\hat{\theta}_i \to \theta_*$ a.s.-$\mathbb{P}_{\mathcal{W}|\mathcal{D}}$, by Proposition 4.3. Here, $\lambda_A = \sup_{x\in\mathcal{X}}\left\|W_1\tilde{A}(x) - \bar{A}\right\|_2^2 < \infty$, by Assumption (A2)(ii) and the boundedness of W.

Now consider the term within the sum in I_2. Without loss of generality, assume $i > j$. Let $\mathcal{F}_j$ denote the natural filtration with respect to the Markov chain $\{X_k\}$, upto index j. Then, we have

$$\mathbb{E}\left[\left\langle\left(W_{i+1}\tilde{A}(X_{i+1}) - \bar{A}\right)\left(\hat{\theta}_i - \theta_*\right), \left(W_{j+1}\tilde{A}(X_{j+1}) - \bar{A}\right)\left(\hat{\theta}_j - \theta_*\right)\right\rangle\right]$$
$$\leq \frac{R_\theta^2}{i^\gamma j^\gamma}\mathbb{E}\left[\left\langle W_{i+1}\tilde{A}(X_{i+1}) - \bar{A}, W_{j+1}\tilde{A}(X_{j+1}) - \bar{A}\right\rangle\right]$$
$$= \frac{R_\theta^2}{i^\gamma j^\gamma}\mathbb{E}\left[\mathbb{E}\left[\langle W_{i+1}\tilde{A}(X_{i+1}) - \bar{A}, W_{j+1}\tilde{A}(X_{j+1}) - \bar{A}\rangle \,|\mathcal{F}_{j+1}\right]\right]$$
$$= \frac{R_\theta^2}{i^\gamma j^\gamma}\mathbb{E}\left[\left\langle\mathbb{E}\left[W_{i+1}\tilde{A}(X_{i+1}\,|\mathcal{F}_{j+1}\right] - \bar{A}, W_{j+1}\tilde{A}(X_{j+1}) - \bar{A}\right\rangle\right],$$

where the first step uses $\left\|\hat{\theta}_i - \theta_*\right\| \leq R_\theta i^{-\gamma}$, for some $\gamma \in (0, \eta - 1/2)$ and $R_\theta < \infty$, by Proposition 4.3, while the second step follows from the tower property, conditioning on the filtration $\mathcal{F}_{j+1}$.

Proceeding from here, we have

$$\frac{R_\theta^2}{i^\gamma j^\gamma}\mathbb{E}\left[\left\langle\mathbb{E}\left[W_{i+1}\tilde{A}(X_{i+1})\,|\mathcal{F}_{j+1}\right] - \bar{A}, W_{j+1}\tilde{A}(X_{j+1}) - \bar{A}\right\rangle\right]$$
$$\leq \frac{R_\theta^2}{i^\gamma j^\gamma}\mathbb{E}\left[\left\|\mathbb{E}\left[W_{i+1}\tilde{A}(X_{i+1})\,|\mathcal{F}_{j+1}\right] - \bar{A}\right\|_F\left\|W_{j+1}\tilde{A}(X_{j+1}) - \bar{A}\right\|_F\right]$$
$$\leq \frac{R_\theta^2}{i^\gamma j^\gamma}\mathbb{E}\left[\left\|\mathbb{E}\left[W_{i+1}A(X_{i+1})\,|\mathcal{F}_{j+1}\right] - \bar{A}\right\|_F\left(\|W_{j+1}A(X_{j+1})\|_F + \left\|\bar{A}\right\|_F\right)\right]$$
$$\leq \frac{(1+W_{\max})A_{\max}R_\theta^2}{i^\gamma j^\gamma}\mathbb{E}\left[\left\|\mathbb{E}\left[W_{i+1}A(X_{i+1})\,|\mathcal{F}_{j+1}\right] - \bar{A}\right\|_F\right]$$
$$\leq (1+W_{\max})A_{\max}^2R_\theta^2M\frac{\kappa^{i-j}}{i^\gamma j^\gamma}.$$

We first bound the inner product using Frobenius norms. In the third step, we bound the second term within the expectation using Assumption (A2)(ii) and the boundedness of W_j. The final step follows from Lemma S2.2.

So far, we have shown that

$$I_2 \leq (1 + W_{\max}) A_{\max}^2 R_\theta^2 M \cdot \frac{1}{t} \sum_{i \neq j}^{t} \frac{\kappa^{i-j}}{i^\gamma j^\gamma}.$$

Consider the double sum above. Grouping terms by $l = |i - j|$, we have

$$\sum_{i \neq j}^{t} \frac{\kappa^{i-j}}{i^\gamma j^\gamma} = 2 \sum_{l=1}^{t-1} S_{t,l} \kappa^l, \quad \text{where} \quad S_{t,l} = \sum_{j=1}^{t-l} \frac{1}{j^\gamma (j+l)^\gamma}.$$

Then $S_{t,l} \leq \sum_{j=1}^{t-l} \frac{1}{j^{2\gamma}}$. For any fixed l, $\lim_{t\to\infty} \sum_{j=1}^{t-l} \frac{1}{j^{1+2\gamma}} < \infty$, and so $\lim_{t\to\infty} \frac{1}{t} \sum_{j=1}^{t-l} \frac{1}{j^{2\gamma}} = 0$, by Kronecker's lemma. Hence, $S_{t,l}/t \to 0$, as $t \to \infty$. Then, by the Dominated Convergence Theorem, we have $\lim_{t\to\infty} \frac{1}{t} \sum_{l=1}^{t-1} S_{t,l} \kappa^l = 0$. It follows that $I_2 \to 0$ as $t \to \infty$, and so $\frac{1}{\sqrt{t}} \sum_{i=1}^{t} (W_{i+1} \tilde{A}(X_{i+1}) - \bar{A})(\hat{\theta}_i - \theta_*) = o_p(1)$. This concludes the proof. □

The following is a restatement of Lemma 2.11 from van der Vaart (1998).

Lemma S2.3. *Suppose that $X_n \implies X$ for a random vector X with a continuous distribution function. Then* $\sup_x |P(X_n \leq x) - P(X \leq x)| \to 0$.

Proof of Theorem 4.2

Let $f(x) = \tilde{A}(x)\theta_* - \tilde{b}(x)$. Then, by Assumption (A2)(ii), f is bounded, and

$$\lim_{t\to\infty} \mathbb{E}[f(X_t)] = \bar{A}\theta_* - \bar{b} = 0$$

under the stationary distribution μ.

By the Poisson equation (see e.g., Douc et al. (2018)), there exists a bounded function u such that

$$u(x) - \mathcal{P}u(x) = f(x).$$

For $t \geq 0$, we define the following terms:

$$e_{t+1} = u(X_{t+1}) - \mathcal{P}u(X_t),$$
$$r_{t+1} = \mathcal{P}u(X_t) - \mathcal{P}u(X_{t+1}).$$

Let $\mathcal{F}_t = \sigma(\{X_i\}_{i=1}^t)$ denote the natural filtration induced by the Markov chain $\{X_t\}$. Then $f(X_t) = e_t + r_t$, where e_t is a martingale difference sequence, since

$$\mathbb{E}[e_{t+1}|\mathcal{F}_t] = \mathbb{E}[u(X_{t+1}) \mid \mathcal{F}_t] - \mathcal{P}u(X_t) = 0,$$

and

$$\frac{1}{\sqrt{t}} \sum_{i=1}^{t} r_i = \frac{1}{\sqrt{t}} (\mathcal{P}u(X_0) - \mathcal{P}u(X_t)) \to 0 \quad a.s., \tag{S2.3}$$

as $t \to \infty$, by a telescoping sum argument. Then from (4.6) we have

$$\begin{aligned} \sqrt{t}(\bar{\theta}_t - \theta_*) &= -\frac{1}{\sqrt{t}} \bar{A}^{-1} \sum_{i=1}^{t} f(X_{i+1}) + o_p(1) \\ &= -\frac{1}{\sqrt{t}} \bar{A}^{-1} \sum_{i=1}^{t} e_{i+1} + o_p(1), \end{aligned} \tag{S2.4}$$

by (S2.3). Combined with Proposition 4.2, this implies that

$$\frac{1}{\sqrt{t}} \bar{A}^{-1} \sum_{i=1}^{t} e_{i+1} \implies \mathcal{N}(0, \bar{A}^{-1} Q (\bar{A}^{-1})^{\mathsf{T}}). \tag{S2.5}$$

On the other hand, since e_{i+1} is uniformly bounded (as $f(x)$ is uniformly bounded for all $x \in \mathcal{X}$), the Lindenberg condition is satisfied, that is,

$$\sum_{i=1}^{t} \mathbb{E}\left[\frac{\|e_i\|_2^2}{t} I_{\{\|e_i\|_2/\sqrt{t} \geq \epsilon\}} \,|\mathcal{F}_{i-1} \right] \to 0,$$

in probability, as $t \to \infty$. So, by the martingale central limit theorem (e.g., Lemma A.3. of Liang (2010)), we have

$$\frac{1}{\sqrt{t}} \bar{A}^{-1} \sum_{i=1}^{t} e_{i+1} \implies \mathcal{N}(0, \Lambda), \tag{S2.6}$$

where Λ is a positive definite matrix with

$$\bar{A}^{-1}\sum_{i=1}^{t}\mathbb{E}[e_i e_i^{\mathsf{T}}/t|\mathcal{F}_{i-1}]\left(\bar{A}^{-1}\right)^{\mathsf{T}} \to \Lambda, \tag{S2.7}$$

in probability as $t \to \infty$. It follows from (S2.5) and (S2.6) that $\Lambda = \bar{A}^{-1}Q\left(\bar{A}^{-1}\right)^{\mathsf{T}}$, and so, by (S2.7), we have

$$\sum_{i=1}^{t}\mathbb{E}[e_i e_i^{\mathsf{T}}/t|\mathcal{F}_{i-1}] \to Q, \tag{S2.8}$$

in probability, as $t \to \infty$.

Next, from (4.7), we have

$$\begin{aligned}\sqrt{t}(\bar{\hat{\theta}}_t - \bar{\theta}_t) &= -\frac{1}{\sqrt{t}}\bar{A}^{-1}\sum_{i=1}^{t}(W_{i+1}-1)f(X_{i+1}) + o_p(1)\\ &= -\frac{1}{\sqrt{t}}\bar{A}^{-1}\sum_{i=1}^{t}(W_{i+1}-1)e_{i+1} + o_p(1),\end{aligned}$$

using (S2.3). Let $\xi_t = (W_t - 1)e_t$. Then ξ_t is a martingale difference sequence, since

$$\mathbb{E}[\xi_{t+1} \mid \mathcal{F}_t] = \mathbb{E}[W_{t+1}-1]\mathbb{E}[e_{t+1} \mid \mathcal{F}_t] = 0.$$

Since ξ_t is uniformly bounded, the Lindenberg condition holds. Then, by the martingale central limit theorem, conditional on the data $\mathcal{D}$, the term $\frac{1}{\sqrt{t}}\bar{A}^{-1}\sum_{i=1}^{t}\xi_{i+1}$ is asymptotically normal with mean 0 and variance

$$\begin{aligned}\underset{t\to\infty}{p\text{-lim}}\,\bar{A}^{-1}\sum_{i=1}^{t}\mathbb{E}[\xi_i\xi_i^{\mathsf{T}}/t|\mathcal{F}_{i-1}]\left(\bar{A}^{-1}\right)^{\mathsf{T}} &= \underset{t\to\infty}{p\text{-lim}}\,\bar{A}^{-1}\mathrm{Var}(W_1)\sum_{i=1}^{t}\mathbb{E}[e_i e_i^{\mathsf{T}}/t|\mathcal{F}_{i-1}]\left(\bar{A}^{-1}\right)^{\mathsf{T}}\\ &= \bar{A}^{-1}Q\left(\bar{A}^{-1}\right)^{-1},\end{aligned}$$

where the first equality follows by independence of W_i and e_i and the fact that the W_i's are i.i.d., while the second equality follows from (S2.8) and $\mathrm{Var}(W_1) = 1$. So, we have

$$\sqrt{t}(\bar{\hat{\theta}}_t - \bar{\theta}) = -\frac{1}{\sqrt{t}}\bar{A}^{-1}\sum_{i=1}^{t}\xi_{i+1} + o_p(1) \implies \mathcal{N}(0, \bar{A}^{-1}Q(\bar{A}^{-1})^{\mathsf{T}}), \tag{S2.9}$$

as $t \to \infty$, where the asymptotic normality holds conditional on data $\mathcal{D}$.

Let $X \sim \mathcal{N}(0, \bar{A}^{-1}Q(\bar{A}^{-1})^\intercal)$ denote the random variable with the limiting distribution of $\sqrt{t}(\bar{\theta}_t - \theta_*)$ as $t \to \infty$. Applying Lemma S2.3 to the result of Proposition 4.2 and equation (S2.9), respectively, we get

$$\sup_{v \in \mathbb{R}^d} \left| \mathbb{P}_{\mathcal{D}}(\sqrt{t}(\bar{\theta}_t - \theta_*) \leq v) - \mathbb{P}(X \leq v) \right| \to 0, \text{and}$$
$$\sup_{v \in \mathbb{R}^d} \left| \mathbb{P}_{\mathcal{W}|\mathcal{D}}(\sqrt{t}(\bar{\tilde{\theta}}_t - \theta_*) \leq v) - \mathbb{P}(X \leq v) \right| \to 0, \text{in probability},$$

as $t \to \infty$. Then

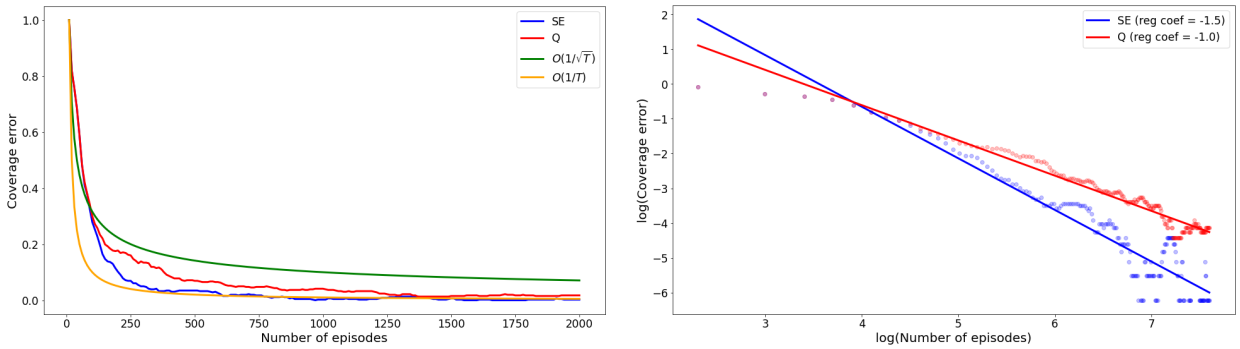
$$\begin{aligned}
&\sup_{v \in \mathbb{R}^d} \left| \mathbb{P}_{\mathcal{W}|\mathcal{D}}(\sqrt{t}(\bar{\tilde{\theta}}_t - \theta_*) \leq v) - \mathbb{P}_{\mathcal{D}}(\sqrt{t}(\bar{\theta}_t - \theta_*) \leq v) \right| \\
&\leq \sup_{v \in \mathbb{R}^d} \left| \mathbb{P}_{\mathcal{D}}(\sqrt{t}(\bar{\theta}_t - \theta_*) \leq v) - \mathbb{P}(X \leq v) \right| \\
&+ \sup_{v \in \mathbb{R}^d} \left| \mathbb{P}_{\mathcal{W}|\mathcal{D}}(\sqrt{t}(\bar{\tilde{\theta}}_t - \theta_*) \leq v) - \mathbb{P}(X \leq v) \right| \\
&\to 0,
\end{aligned}$$

in probability, as $t \to \infty$. □

S3 Additional Experiments

In this section, we provide the study of the second-order accuracy of our bootstrap method in the Frozenlake environment considered in Section 5.1.

To empirically evaluate the second-order accuracy, we measured the coverage error rates of the 95% confidence intervals for the value function of the initial state in the Frozenlake environment. We use TD learning to estimate the value function, and the quantile and standard error estimators, computed from the online bootstrap estimates, to generate the confidence intervals. Figure 8a shows the empirical coverage errors of the quantile and standard error estimators as a function of the number of episodes in the RL Frozenlake environment. The rates are re-scaled to start from 1 at the first time step. In both cases, we can see that the coverage error decreases at a rate faster than $O(1/\sqrt{t})$ initially, and eventually reaches a rate of $O(1/t)$ or better.



(a) Coverage error rates

(b) Least squares estimates

Figure 8: Figure 8a shows the coverage error rates for the quantile and SE confidence intervals, and Figure 8b shows the linear regression coefficients for the log coverage error rates against the log number of episodes.

We then computed estimates of the coverage error rate by regression the log of the coverage errors against the log of the number of episodes. We would expect a first-order accurate method to have a regression coefficient of $-1/2$ or lower (corresponding to a coverage error rate of $O(1/\sqrt{t})$), while a second-order accurate method would have a coefficient of -1 or lower. As shown in Figure 8b, both the quantile and standard error have regression coefficients of -1 or lower, which demonstrates that they both achieved second-order accuracy.