ON THE PERFORMANCE OF TEMPORAL DIFFERENCE LEARNING WITH NEURAL NETWORKS

Anonymous authors
Paper under double-blind review

ABSTRACT

Neural Temporal Difference (TD) Learning is an approximate temporal difference method for policy evaluation that uses a neural network for function approximation. Analysis of Neural TD Learning has proven to be challenging. In this paper we provide a convergence analysis of Neural TD Learning with a projection onto $B(\theta_0, \omega)$, a ball of fixed radius ω around the initial point θ_0. We show an approximation bound of $O(\epsilon + 1/\sqrt{m})$ where ϵ is the approximation quality of the best neural network in $B(\theta_0, \omega)$ and m is the width of all hidden layers in the network.

1 INTRODUCTION

Temporal difference (TD) learning is considered to be a major milestone of reinforcement learning (RL). Proposed by Sutton (1988), TD Learning uses the Bellman error, which is a difference between an agent's predictions in a Markov Decision Process (MDP) and what it actually observes, to drive the process of learning an estimate of the value of every state.

To deal with large state-spaces, TD learning with linear function approximation was introduced in (Tesauro et al. (1995)). A mathematical analysis was given in (Tsitsiklis & Van Roy (1996)), which shows the process converges under appropriate assumptions on step-size and sampling procedure. However, with nonlinear function approximation, TD Learning is not guaranteed to converge, as observed in Tsitsiklis & Van Roy (1996) (see also (Achiam et al. (2019) for a more recent treatment).

Nevertheless, TD with neural network approximation, referred to as Neural TD, is used in practice despite the lack of strong theoretical guarantees. To our knowledge, rigorous analysis of the convergence of Neural TD was only addressed in the three papers Cai et al. (2019); Xu & Gu (2020); Cayci et al. (2021).

In Cai et al. (2019), a single hidden layer neural architecture was considered along with projection on a ball around the initial condition; approximate convergence was proved to an approximate stationary point of a certain function related to the linearization around the initial point. This result was generalized to multiple hidden layers in Xu & Gu (2020), but this generalization required projection on a ball of radius of $\omega \sim m^{-1/2}$ around the initial point, where m is the width of the hidden layers. Because the radius of of this projection goes to zero with m, this effectively fixes the neural network to a small distance from its initial condition. Both Cai et al. (2019); Xu & Gu (2020) additionally required certain regularity conditions on the policy. Finally, Cayci et al. (2021) gave a convergence result for a single hidden layer, also with a projection onto a radius of $\omega \sim m^{-1/2}$ around the initial point, but with the final objective being the representation error of neural approximation without any kind of linearization. This result also required a condition on the representability of the value function of the policy in terms of features from a random initialization.

In this paper, we analyze Neural TD with a projection with a projection onto $B(\theta_0, \omega)$, a ball of fixed radius ω around the initial point θ_0. We show an approximation bound of $O(\epsilon + 1/\sqrt{m})$ where ϵ is the approximation quality of the best neural network in $B(\theta_0, \omega)$. Our result improves on previous work because it does not require taking the radius ω to decay with m, does not make any regularity or representability assumptions on the policy, applies to any number of hidden layers, and bounds the error associated with the neural approximation without any kind of linearization around the initial condition.

The main technical difference between our paper and previous works is the choice of norm for analysis. We will describe this at a more technical level in main body of the paper, but we use a norm introduced by Ollivier (2018), which is a convex combination of the usual l_2-norm weighted by the stationary distribution of the policy with the so-called Dirichlet seminorm. The Dirichlet seminorm has previously been used in the convergence analysis of Markov chains (Diaconis & Saloff-Coste (1996); Levin & Peres (2017)). Remarkably, it was shown in Ollivier (2018) that neural TD is exactly gradient descent on this convex combination of norms if the underlying policy is reversible.

In the case where the policy is not reversible, these results were partially generalized in Liu & Olshevsky (2021), where it was shown that TD with linear function approximation can be viewed as a so-called *gradient splitting*, a process which is analogous to gradient descent. We build heavily on that interpretation here. Our main technical argument is that neural TD approximates the gradient splitting process at each step, so that despite the nonlinearity of the approximation, an improvement in approximation quality can be guaranteed unless the system is already close to the best possible approximation over the projection radius. Notably, our arguments do not imply that the neural network stays close to its initialization and our empirical simulations show a significant benefit from taking the projection radius ω *not* to decay with the width m.

2 PRELIMINARIES

2.1 MARKOV DECISION PROCESSES

In this section, we will introduce MDP. While all this material is standard (e.g., Sutton & Barto (2018)), it is necessary to standardize notation and make our presentation self-contained.

A finite discounted-reward MDP can be described by a tuple $(S, A, P_{\text{env}}, r, \gamma)$, where $S = \{s_1, s_2, \ldots, s_n\}$ is a finite state-space whose elements are vectors; A is a finite action space; $P_{\text{env}} = (P_{\text{env}}(s'|s,a))_{s,s' \in S, a \in A}$ is the transition probability matrix, where $P_{\text{env}}(s'|s,a)$ is the probability of transitioning from s to s' after taking action a; $r : S \times A \to R$ is the reward function; and $\gamma \in (0,1)$ is the discount factor. A policy π in an MDP is a mapping $\pi : S \times A \to [0,1]$ such that $\sum_{a \in A} \pi(s,a) = 1$ for all $s \in S$. Here, $\pi(s,a)$ means the probability that the agent takes action a in state s.

Given a policy π, we define the corresponding transition matrix from state to state as $P = (P(s'|s))_{s,s' \in S}$ where

$$P(s'|s) = \sum_{a \in A} \pi(s,a) P_{\text{env}}(s'|s,a).$$

We also define the corresponding reward function as

$$r(s) = \sum_{a} pi(s,a) r(s,a).$$

P and $r(s)$ naturally depends on the policy π but throughout this paper the policy will be fixed and hence we will suppress this dependence in the notation.

The stationary distribution μ corresponding to the policy π is defined to be a non negative vector with coordinates summing to one satisfying $\mu^T = \mu^T P$. The Perron-Frobenius theorem guarantees that such a μ exists and is unique subject to some conditions on P, e.g., aperiodicity and irreducibility (Gantmacher (1964)). We use $\mu(s)$ to denote each entry of μ. The value function of the policy π is defined as:

$$V^*(s) = \mathbb{E}_{s,a \sim \pi} \left[\sum_{t=0}^{+\infty} \gamma^t r(s^t) \right],$$

where $\mathbb{E}_{s,a \sim \pi}$ stands for the expectation when the starting state is s and actions are taken from the policy π, and s^t is the t'th state encountered. Note that this quantity depends on π, but once again we suppress this dependence because the policy will be fixed throughout this paper. Moreover, note that, despite the star superscript, V^* is not the optimal value function but rather the true value function corresponding to the policy π.

This value function satisfies satisfies the Bellman equation,

$$V^* = R + \gamma P V^* \tag{1}$$

where V^* is a vector whose i'th element is $V^*(s_i)$ and R is a vector whose i'th element $R(s_i) = r(s_i)$ is the instantaneous reward in a given state s_i.

Throughout the paper, we will assume the following:

Assumption 2.1. *Suppose for any $s, a \in S \times A$, the function $|r(s,a)| \leq r_{\max}$.*

This immediately implies that

$$|V^*(s)| \leq \frac{r_{\max}}{1-\gamma}, \ \forall s \in S. \tag{2}$$

2.2 Markov Chain Noise Model

There are two standard sampling models. The simplest model involves i.i.d. sampling of the states s_t at each step from the stationary distribution μ.

Alternatively, in the Markov model the states s_t are collected from a single path of Markov chain transitioning according to P. It is still assumed in this case that the initial distribution is μ, so that the distribution of each s_t is still μ (with the difference that the successive states are now correlated). This can always be approximately satisfied by generating a sufficiently long path from P and ignoring the initial states.

Assuming that the underlying Markov chain mixes with a geometric rate is common in many analyses, like Bhandari et al. (2018); Liu & Olshevsky (2021). Here, we make the following assumption:

Assumption 2.2. *There exists constant $C > 0$ and $\beta \in (0,1)$ such that*

$$\max_s ||P^t_{s,:} - \mu||_{\mathrm{TV}} \leq C\beta^t$$

where P^t means matrix P to the power of t, $P^t_{s,:}$ stands for the row of P^t corresponding to state s, and $||\cdot||_{\mathrm{TV}}$ stands for the total variance.

This assumption guarantees "mixing": no matter the initial distribution, the state will get closer and closer to the stationary distribution μ as t increases. As pointed out in Levin & Peres (2017), this assumption always holds when the Markov chain is irreducible and aperiodic. Another useful quantity is called mixing time, $\tau_{\mathrm{mix}}(\epsilon_{\mathrm{mix}})$. It is defined to be the smallest integer t such that

$$\max_s ||P^t_{s,:} - \mu||_{\mathrm{TV}} \leq \epsilon_{\mathrm{mix}}.$$

Note that Assumption 2.2 implies

$$\tau_{\mathrm{mix}}(\epsilon_{\mathrm{mix}}) = \log_\beta \frac{\epsilon_{\mathrm{mix}}}{C}. \tag{3}$$

For simplicity, we will use τ_{mix} without specifying its dependence on ϵ_{mix} throughout the paper.

2.3 D-norm and Dirichlet Norm in MDPs

We now introduce the so-called D-norm $||\cdot||_D$ and the Dirichlet semi-norm $||\cdot||_{\mathrm{Dir}}$ associated with a policy. While the former has long been used for analysis of temporal difference learning dating back to Tsitsiklis & Van Roy (1996), the latter has, to our knowledge, been introduced in the context of RL relatively recently in Ollivier (2018).

Let $D = \mathrm{diag}(\mu(s))$ be the diagonal matrix whose elements are given by the entries of the stationary distribution μ. Given a function $f : S \to R$, its D-norm is defined as

$$||f||_D^2 = f^T D f = \sum_{s \in S} \mu(s) f(s)^2. \tag{4}$$

The D-norm is similar to the Euclidean norm except each entry is weighted proportionally to the stationary distribution. We also define the Dirichlet semi-norm of f:

$$||f||_{\mathrm{Dir}}^2 = \frac{1}{2} \sum_{s,s' \in S} \mu(s) P(s'|s)(f(s') - f(s))^2. \tag{5}$$

Note that $||f||_{\text{Dir}}$ depends on the policy both through the stationary distribution $\mu(s)$ as well as through the transition matrix P.

Finally, following Ollivier (2018) the weighted combination of the D-norm and the Dirichlet semi-norm is denoted as $\mathcal{N}(f)$ and defined as

$$\mathcal{N}(f) = (1-\gamma)||f||_D^2 + \gamma||f||_{\text{Dir}}^2.$$

2.4 Neural Network Based Approximation

In this section we closely follow the notation from the previous works in Cai et al. (2019); Allen-Zhu et al. (2019); Liu et al. (2020); Ollivier (2018) on neural approximations in reinforcement learning. We define multi-layer fully connected neural network is given by the following recursion:

$$x^{(k)} = \frac{1}{\sqrt{m}}\sigma(\theta^{(k)}x^{(k-1)}), \text{ for } k \in \{1, \dots, K\}$$

where σ is and activation function and the input is one of the states of the MDP:

$$x^{(0)} = s \in S.$$

We then define

$$V(s, \theta) = \frac{1}{\sqrt{m}} b^T x^{(K)},$$

to be the output with no activation function on the output. Each entry of b satisfies $|b_r| \leq 1, \forall r$. We assume that all the hidden layers have the same width which we denote by m, i.e., all the matrices $\theta^{(k)}$ have first dimension of m. Note that the total number of layers in the neural network is denoted by K here.

We will stack up the weights of different layers as $\theta = [\theta^{(1)}, \dots, \theta^{(K)}]$, and the norm of this stacked-up vector is defined as

$$||\theta||^2 = \sum_{k=1}^{K} ||\theta^{(k)}||_F^2$$

where $||\cdot||_F$ is Frobenius norm. During the training process, only the weights θ will be updated while the final weights b will be left to their initial value.

We can view this neural network as mapping the parameters θ to a vector with as many entries as the number of states. Specifically, we can define the vector $V(\theta)$ whose i'th entry is

$$[V(\theta)]_i = V(s_i, \theta), \text{for all } i \in 1, 2, \dots, n.$$

Note that n in the above equation denotes the number of states in the MDP, i.e., $|S|$. The Jacobian of $V(\theta)$ is then the matrix

$$\nabla_\theta V(\theta) = \begin{bmatrix} \nabla_\theta V(s_1, \theta) \\ \vdots \\ \nabla_\theta V(s_n, \theta) \end{bmatrix}.$$

Here, for an abuse of notation, θ is stacked up to be a column vector and $\nabla_\theta V(s, \theta)$ is a row vector.

A standard assumption made to simplify the analysis is a uniform upper bound on the norm of the states:

Assumption 2.3. *Suppose for any $i \in \{1, 2, \dots, n\}$, $||s_i|| \leq 1$ where $||\cdot||$ stands for the l_2-norm.*

Assumption 2.3 can always be satisfied by scaling, as we typically have control of how we choose to represent the states of the MDP. It is also a common assumption which appears in many previous works (e.g., Cai et al. (2019); Allen-Zhu et al. (2019)).

We will assume the activation function σ satisfies a number of smoothness properties, which we will next define. A differentiable function $f : \mathbb{R} \to \mathbb{R}$ is H-smooth if

$$|f'(x) - f'(y)| \leq H|x - y|, \ \forall x, y$$

where f' stands for the derivative of f.

A function f is called L-Lipschitz if

$$|f(x) - f(y)| \leq L|x - y|, \ \forall x, y.$$

We will also find it convenient to use smooth and Lipschitz activation functions.

Assumption 2.4. *The activation function σ is smooth and l-Lipschitz.*

2.5 NEURAL TD

In this section, we introduce (projected) neural TD learning. At each time step t, this algorithm samples a state s from the stationary distribution from either of two sampling models discussed earlier, generates the next state s' in the MDP, and computes the temporal difference error, defined as

$$\delta_t = r(s) + \gamma V(s', \theta_t) - V(s, \theta_t).$$

Defining $g(\theta_t)$ as

$$g(\theta_t) = \nabla_\theta V(s, \theta_t)\delta_t,$$

neural TD updates the weights as:

$$\begin{aligned} \theta_{t+0.5} &= \theta_t + \alpha_t g(\theta_t) \\ \theta_{t+1} &= \mathbf{Proj}(\theta_{t+0.5}) \end{aligned} \tag{6}$$

where the projection is onto a ball of radius ω around the initial condition:

$$\mathbf{Proj}(\theta) = \arg \min_{x: ||x - \theta_0|| \leq \omega} ||x - \theta||. \tag{7}$$

Projection is a common tool in neural TD and neural Q-learning to try to stabilize the iterates, since divergence can occur Achiam et al. (2019); Van Hasselt et al. (2018). Most analyses of TD learning proceed by comparing the evolution to the mean-path update, defined as

$$\begin{aligned} \theta_{t+0.5} &= \theta_t + \alpha_t \bar{g}(\theta_t) \\ \theta_{t+1} &= \mathbf{Proj}(\theta_{t+0.5}) \end{aligned}$$

where

$$\begin{aligned} \bar{g}(\theta_t) &= \mathbb{E}[g(\theta_t)|\theta_t] \\ &= \sum_s \mu(s)\nabla_\theta V(s, \theta_t)\mathbb{E}_{s'|s}[r(s) + \gamma V(s', \theta_t) - V(s, \theta_t)] \\ &= \nabla_\theta V(\theta_t)^T D(R + \gamma P V(\theta_t) - V(\theta_t)). \end{aligned}$$

It is convenient to rewrite $\bar{g}(\theta_t)$ in terms of the difference between $V(\theta_t)$ and V^*, which can be done by subtracting Eq.(1) with the result being

$$\bar{g}(\theta_t) = \nabla_\theta V(\theta_t)^T D(\gamma P - I)(V(\theta_t) - V^*). \tag{8}$$

For simplicity, for the rest of paper we will define Θ to be the set onto which we are projecting:

$$\Theta = \{\theta \mid ||\theta - \theta_0|| \leq \omega\}.$$

Finally, we will say that $V(\hat{\theta}_*)$ or $\hat{V}^*$ *is an ϵ-approximation to V^* if*

$$\max_{s \in S} |V(s, \hat{\theta}_*) - V^*(s)| \leq \epsilon. \tag{9}$$

Using Eq.(2), this definition immediately implies that

$$|V(s, \hat{\theta}_*)| \leq \epsilon + \frac{r_{\max}}{1 - \gamma}, \ \forall s \in S. \tag{10}$$

3 OUR MAIN RESULT

We can now state the main contribution of this paper, which is a performance result for Neural TD. We will require an assumption to the effect that the initialization is not too large. Recall that m is the width of each layer.

Assumption 3.1. *Assume that* $||\theta_0^{(k)}|| \leq O(\sqrt{m})$ *holds for all* $k \in \{1, 2, \ldots, K\}$.

This is a common assumption; theoretical justification can be found in Liu et al. (2020). In particular, this assumption holds with high probability for a random Gaussian initialization if, for example, K (the depth) grows slower than exponentially in m (the width).

Now we are ready to state our result.

Theorem 3.1. *Suppose Assumptions 2.3 and 2.4 and 3.1 hold, and suppose* θ_t *is generated by projected Neural TD with the constant step-size* $\alpha_t = \alpha$. *We further assume there is some* $\hat{\theta}_* \in B(\theta_0, \omega)$ *which is an* ϵ*-approximation to* V^*.

(a) Under i.i.d. sampling, we have

$$\frac{1}{T}\sum_{t=0}^{T-1} \mathbb{E}[\mathcal{N}(V(\theta_t) - V(\hat{\theta}_*))] \leq \frac{||\theta_0 - \hat{\theta}_*||^2}{2\alpha T} + O\left(\epsilon + \frac{1}{\sqrt{m}}\right) + O\left(\alpha(1 + \epsilon + \epsilon^2)\right).$$

In particular, if $\alpha = T^{-1/2}$ *and* $\epsilon \leq 1$, *we have*

$$\frac{1}{T}\sum_{t=0}^{T-1} \mathbb{E}[\mathcal{N}(V(\theta_t) - V(\hat{\theta}_*))] \leq \frac{||\theta_0 - \hat{\theta}_*||^2}{2\sqrt{T}} + O\left(\epsilon + \frac{1}{\sqrt{m}}\right) + O\left(\frac{1}{\sqrt{T}}\right).$$

(b) Under Markov sampling, we have

$$\begin{aligned}\frac{1}{T}\sum_{t=0}^{T-1} \mathbb{E}[\mathcal{N}(V(\theta_t) - V(\hat{\theta}_*))] \quad &\leq \quad \frac{||\theta_0 - \hat{\theta}_*||^2}{2\alpha T} + O\left(\epsilon + \frac{1}{\sqrt{m}}\right) + O\left(\alpha(1 + \epsilon + \epsilon^2)\right) \\ &\quad + O\left((1+\epsilon)\alpha \frac{1}{1-\beta} \log \frac{C}{\alpha}\right),\end{aligned}$$

where C, β *come from the Assumption 2.2 on the mixing time.*

In particular, if $\alpha = T^{-1/2}$ *and* $\epsilon \leq 1$, *we have*

$$\begin{aligned}\frac{1}{T}\sum_{t=0}^{T-1} \mathbb{E}[\mathcal{N}(V(\theta_t) - V(\hat{\theta}_*))] \quad &\leq \quad \frac{||\theta_0 - \hat{\theta}_*||^2}{2\sqrt{T}} + O\left(\epsilon + \frac{1}{\sqrt{m}}\right) \\ &\quad + O\left(\frac{1}{1-\beta} \frac{\log(C\sqrt{T})}{\sqrt{T}}\right).\end{aligned}$$

As mentioned earlier, the key distinguishing feature of this theorem is the choice of norm. The left-hand side of all the equations measures the difference between $V(\theta_t)$ and the best possible $V(\hat{\theta}_*)$ within $B(\theta_0, \omega)$ by taking the $\mathcal{N}(\cdot)$ norm.

We note that since, trivially,

$$||f||_D^2 \leq \frac{\mathcal{N}(f)}{1-\gamma}$$

where γ is defined to be the discount factor, one can simply replace the left-hand sides of all the equations by $||f||_D^2$ to obtain results that look more similar to the previous literature on TD, which usually proceeds based on an analysis in the D-norm (e.g., Tsitsiklis & Van Roy (1996)).

As is common in analyses of SGD, the performance measure is the average of performance measures from 1 to T. If a particular $\theta_{t'}$ is sought that satisfies the bounds obtained, a standard trick is to choose t' to be uniform from $1, \ldots, T$. In that case, $E[\mathcal{N}(V(\theta_{t'}) - V(\hat{\theta}_*)]$ is exactly the left-hand side of all the bounds in the theorem.

Looking at the right-hand sides of all the equations, the theorem guarantees a final error of $O(\epsilon + 1/\sqrt{m})$, with a convergence rate that scales as $\tilde{O}(1/\sqrt{T})$. The difference between the two cases is that that the Markov sampling case contains an additional term containing the mixing time. As a result of this extra term, the convergence time in the Markov case is worse by a factor of $\log \sqrt{T}$.

We now revisit the discussion of novelty of this paper. First, in contrast to previous work, we do not assume the projection radius ω has to decay with m, nor do we restrict our analysis to the single hidden layer case. In our proof, the projection radius appears as a constant in the $O(\cdot)$ notation, which is why we need to assume it is a constant. Second, the left-hand side of the equations is a measure of the error $V(\theta_t) - V(\hat{\theta}_*)$, the difference between the error of the neural network and the best possible error in $B(\theta_0, \omega)$. Moreover, as already discussed, the left-hand side of the equations above is greater than the quantity

$$(1-\gamma)||V(\theta_t) - V(\hat{\theta}_*)||_D^2 = (1-\gamma)\sum_s \mu(s)(V(s,\theta_t) - V(s,\hat{\theta}_*))^2,$$

which is a natural measure of the average error. The point is that the network is not being linearized around the initial condition in any sense. Finally, no additional assumptions on the policy are being made here; in particular, no assumptions that the policy is regular or that it is representable are necessary. As outlined in the introduction, out of the four potentially undesirable elements discussed here (small projection radius, linearization around the initial condition, assumptions on policy, restriction to single layer case) all previous papers suffered from at least three.

4 Distinguishing feature of our analysis

The main technical difference between our work and the previous papers is the use of the function $\mathcal{N}(\cdot)$ to measure the approximation error. Here we follow the papers Ollivier (2018); Liu & Olshevsky (2021) which explained why this function is the "right" function to analyze policy evaluation.

In Ollivier (2018) it was shown that if the matrix P corresponds to a reversible Markov chain, then

$$E[\bar{g}(\theta_t)] = \nabla_\theta \mathcal{N}(f).$$

This makes neural TD very easy to analyze, as it can be viewed as gradient descent. Unfortunately, in practice policies obtained in reinforcement learning are almost never reversible.

In Liu & Olshevsky (2021), it was shown how to further use the function $\mathcal{N}(\cdot)$ to analyze TD with linear approximation when the policy is not necessarily reversible. The key idea was the notion of a gradient splitting: a linear function $h(\theta)$ is said to be a gradient splitting of a convex quadratic $f(\theta)$ minimized at $\theta = a$ if

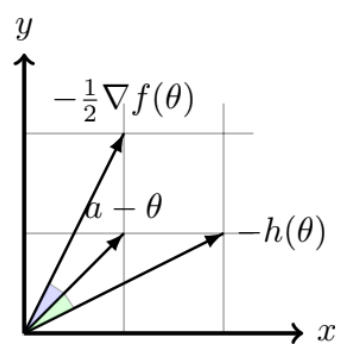


Figure 1: An illustration of the key property of gradient splitting: $h(\theta)$ has the same inner product with $a - \theta$ as $(1/2)\nabla f(\theta)$.

$$\frac{1}{2}\nabla f(\theta)^T(a-\theta) = h(\theta)^T(\theta - a). \tag{11}$$

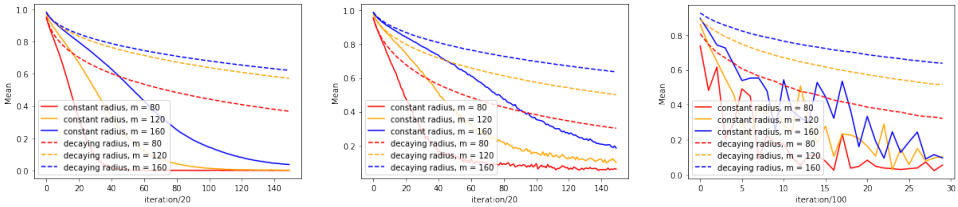
In other words, $h(\theta)$ has exactly the same inner product with the "direction to the optimal solution" as the true gradient of $f(\theta)$ (up to the factor of $1/2$). The significance of this is that it allows many analyses of gradient descent to be modified to analyze TD with linear approximation, since the key step in analyses of gradient descent is usally to argue that the left-hand side of Eq.(11) is negative, signifying that gradient descent "makes progress" towards the optimal solution.

In Liu & Olshevsky (2021) it was shown that for TD with linear approximation, $\bar{g}(\theta)$ is exactly the gradient splitting of $\mathcal{N}(f)$. We build on that idea in this paper as follows. We use recent NTK-style

bounds from Liu et al. (2020) to argue that with increasing width, the neural approximation gets more linear. For finite m, we can then analyze modify existing techniques for analyzing gradient descent with errors in the gradient evaluations to analyze the neural TD update, which we view as *gradient splitting with error in the evaluations.*

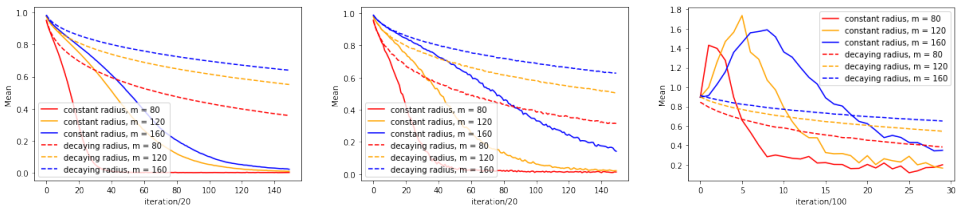
It should be stressed that the most interesting "regime" to which we expect these results to be applicable is when m is only moderately large. Intuitively, when m is too large, the network will be very close to linear, and the benefit over taking random linear features will be small. In terms of our bounds, we would expect the ϵ term in Theorem 3.1 to be large. On the other hand, when m is too small, the error bound scaling with width of $m^{-1/2}$ will not be very attractive. On the other hand, our results can be applied to the "middle range" – say $m \sim 100$ – where, depending on the context of the problem, an $\sim m^{-1/2}$ error is acceptable. Informally, in this regime the network will be sufficiently linear to converge well, but not so linear to lose approximation quality. Our simulations confirm this, as we verify good approximations for networks of approximatley this width on several open AI benchmarks.

5 SIMULATIONS



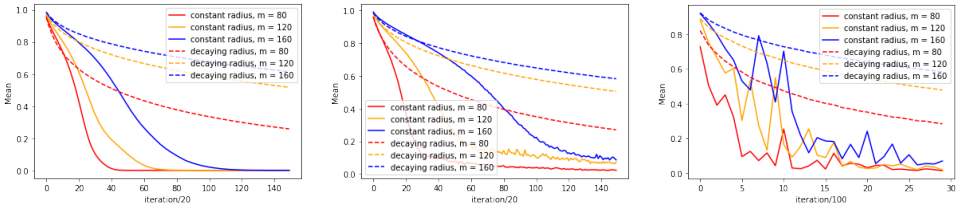
(a) Mountaincar (b) Cartpole (c) Acrobot

Figure 2: Averaged Bellman error using random policy and a three hidden layers neural network



(a) Mountaincar (b) Cartpole (c) Acrobot

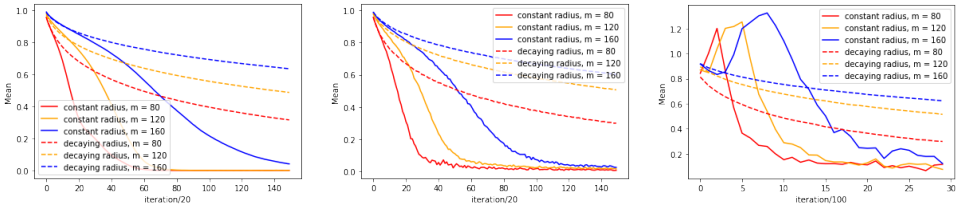
Figure 3: Averaged Bellman error using well-trained policy and a three hidden layers neural network



(a) Mountaincar (b) Cartpole (c) Acrobot

Figure 4: Averaged Bellman error using random policy and a five hidden layers neural network

It might be objected that many analysis of neural network training in the large-width regime proceed by arguing that the neural network stays around its initial point (e.g., Chizat et al. (2019); Telgarsky



(a) Mountaincar (b) Cartpole (c) Acrobot

Figure 5: Averaged Bellman error using well-trained policy and a five hidden layers neural network

(2020)). If so, then the part of our results dealing with avoiding a projection radius that goes to zero with m would not make much of a difference in practice. In these section, we show empirically that, in fact, setting the projection radius ω to be a constant rather than $\omega \sim m^{-1/2}$ does make a substantial difference.

We produce simulations on Open AI Gym tasks: Mountaincar, Cartpole and Acrobot. In each task, either random or well-trained policy is used to sample data to train a fully connected neural network for policy evaluation using Projected Neural TD learning. A well-trained policy in Mountaincar and Cartpole is trained by Proximal policy optimization (PPO) algorithm (Schulman et al. (2017)) while in Acrobot it is trained by Deep Q-Learning (DQN) algorithm (Mnih et al. (2015)). Both algorithms are implemented through the Stable Baselines package Raffin et al. (2021).

We run a total of 3000 steps on each task, computing the Bellman error during training process every 20 (50 in Acrobot task) steps. Different widths m is chosen on each task. In each figure, x-axis shows each time steps we compute Bellman error while y-axis stands for squared Bellman error. It is clear from the figures that networks with a smaller projection radius are significantly outperformed by networks with a constant projection radius.

6 Conclusion

We have provided an analysis of Projected Neural TD for policy evaluation. We have shown that if the projection set is a ball of constant radius $B(\theta_0, \omega)$ around the initial point θ_0, the final approximation error is $O(\epsilon + 1/\sqrt{m})$, where ϵ is the approximation quality of the best neural network in $B(\theta_0, \omega)$ and m is the width of all hidden layers in the network. Our result improves on previous work because it does not require taking the radius ω to decay with m, does not make any regularity or representability assumptions on the policy, applies to any number of hidden layers, and bounds the error associated with the neural approximation without any kind of linearization around the initial condition. We conjecture that unprojected neural TD converges to the optimal solution with high probability provided it begins within a radius of $O(\sqrt{m})$ away from a point which exactly describes the true value function. This conjecture is proved for single hidden layer networks. Finally, we have demonstrated empirically that these changes make a substantial difference in performance, as networks with small projection radius severely underperform on several common benchmarks.

References

Joshua Achiam, Ethan Knight, and Pieter Abbeel. Towards characterizing divergence in deep q-learning. *arXiv preprint arXiv:1903.08894*, 2019.

Zeyuan Allen-Zhu, Yuanzhi Li, and Zhao Song. A convergence theory for deep learning via over-parameterization. In *International Conference on Machine Learning*, pp. 242–252. PMLR, 2019.

Jalaj Bhandari, Daniel Russo, and Raghav Singal. A finite time analysis of temporal difference learning with linear function approximation. In *Conference on learning theory*, pp. 1691–1692. PMLR, 2018.

Qi Cai, Zhuoran Yang, Jason Lee, and Zhaoran Wang. Neural temporal-difference learning converges to global optima. 2019.

Semih Cayci, Siddhartha Satpathi, Niao He, and R Srikant. Sample complexity and overparameterization bounds for temporal difference learning with neural network approximation. *arXiv preprint arXiv:2103.01391*, 2021.

Lenaic Chizat, Edouard Oyallon, and Francis Bach. On lazy training in differentiable programming. *Advances in Neural Information Processing Systems*, 32, 2019.

Persi Diaconis and Laurent Saloff-Coste. Logarithmic sobolev inequalities for finite markov chains. *The Annals of Applied Probability*, 6(3):695–750, 1996.

FR Gantmacher. The theory of matrices. *New York*, 1964.

David A Levin and Yuval Peres. *Markov chains and mixing times*, volume 107. American Mathematical Soc., 2017.

Chaoyue Liu, Libin Zhu, and Misha Belkin. On the linearity of large non-linear models: when and why the tangent kernel is constant. *Advances in Neural Information Processing Systems*, 33: 15954–15964, 2020.

Rui Liu and Alex Olshevsky. Temporal difference learning as gradient splitting. In *International Conference on Machine Learning*, pp. 6905–6913. PMLR, 2021.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533, 2015.

Yann Ollivier. Approximate temporal difference learning is a gradient descent for reversible policies. *arXiv preprint arXiv:1805.00869*, 2018.

Antonin Raffin, Ashley Hill, Adam Gleave, Anssi Kanervisto, Maximilian Ernestus, and Noah Dormann. Stable-baselines3: Reliable reinforcement learning implementations. *Journal of Machine Learning Research*, 22(268):1–8, 2021. URL `http://jmlr.org/papers/v22/20-1364.html`.

Walter Rudin et al. *Principles of mathematical analysis*, volume 3. McGraw-hill New York, 1976.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

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

Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

Matus Telgarsky. Deep learning theory lecture notes, 2020.

Gerald Tesauro et al. Temporal difference learning and td-gammon. *Communications of the ACM*, 38(3):58–68, 1995.

John Tsitsiklis and Benjamin Van Roy. Analysis of temporal-diffference learning with function approximation. *Advances in neural information processing systems*, 9, 1996.

Hado Van Hasselt, Yotam Doron, Florian Strub, Matteo Hessel, Nicolas Sonnerat, and Joseph Modayil. Deep reinforcement learning and the deadly triad. *arXiv preprint arXiv:1812.02648*, 2018.

Pan Xu and Quanquan Gu. A finite-time analysis of q-learning with neural network function approximation. In *International Conference on Machine Learning*, pp. 10555–10565. PMLR, 2020.

A PROOF FOR THE MAIN RESULT

A.1 USEFUL LEMMAS

Before we go into details, we first introduce the following lemmas.

Lemma A.1. *Recall we have defined $\mathcal{N}(f)$ as $\mathcal{N}(f) = (1-\gamma)||f||_D^2 + \gamma||f||_{Dir}^2$. For any function f defined on the state space S, the following equation holds.*

$$-\mathcal{N}(f) = f^T D(\gamma P - I)f.$$

Proof. We can perform the following sequence of manipulations:

$$\begin{aligned}
||f||_{\text{Dir}}^2 &= \frac{1}{2}\sum_{s,s'} \mu(s)P(s'|s)[f(s) - f(s')]^2 \\
&= \frac{1}{2}\sum_{s} \mu(s)f(s)^2 + \frac{1}{2}\sum_{s,s'} \mu(s)P(s'|s)f(s')^2 - \sum_{s,s'} \mu(s)P(s'|s)f(s)f(s') \\
&= \frac{1}{2}\sum_{s} \mu(s)f(s)^2 + \frac{1}{2}\sum_{s'} \mu(s')f(s')^2 - \sum_{s,s'} \mu(s)P(s'|s)f(s)f(s') \\
&= ||f||_D^2 - \sum_{s,s'} \mu(s)P(s'|s)f(s)f(s'),
\end{aligned}$$

where the first equation uses Eq.(5) and the forth equation uses Eq.(4). Thus,

$$\begin{aligned}
f^T D(\gamma P - I)f &= -f^T Df + \gamma f^T D(Pf) \\
&= -||f||_D^2 + \gamma \sum_s \mu(s)f(s) \sum_{s'} P(s'|s)f(s') \\
&= -(1-\gamma)||f||_D^2 - \gamma||f||_{\text{Dir}}^2 \\
&= -\mathcal{N}(f).
\end{aligned}$$

□

Lemma A.2. *(a) Let $h : R \to R$ be any differentiable function. For any $x, y \in R$, there exists $\lambda \in (0,1)$ and $z = \lambda x + (1-\lambda)y$ such that*

$$f(y) - f(x) = f'(z)(y - x).$$

(b) Let $\xi : R^a \to R$ be any differentiable function. For any $x, y \in R^a$, there exists $\lambda \in (0,1)$ and $z = \lambda x + (1-\lambda)y$ such that

$$\xi(y) - \xi(x) = \xi'(z)(y - x).$$

(c) Let $f : R^a \to R^b$ be any differentiable function and $e \in R^b$ be any vector. For any $x, y \in R^a$, there exists $\lambda \in (0,1)$ and $z = \lambda x + (1-\lambda)y$ such that

$$e^T(f(y) - f(x)) = e^T f'(z)(y - x).$$

Proof. (a) This is a directly result of well-known mean value theorem (Theorem 5.10 in Rudin et al. (1976)).

(b) Define $h : R \to R$ such that $h(w) = \xi(x + w(y - x))$. Using the above fact, for any $u, v \in R$, there exists $\lambda \in (0,1)$ such that

$$h(v) - h(u) = h'(\lambda u + (1-\lambda)v)(v - u).$$

By letting $v = 1$ and $u = 0$,

$$\xi(y) - \xi(x) = \xi'(\lambda x + (1-\lambda)y)(y - x).$$

(c) Define $\xi : R^a \to R$ such that $\xi = e^T f$. Using the above fact, for any vectors $x, y \in R^a$,there exists $\lambda \in (0,1)$ and $z = \lambda x + (1-\lambda)y$ such that

$$\xi(y) - \xi(x) = \xi'(z)(y - x).$$

Notice that in this case, $\xi(y) - \xi(x) = e^T(f(y) - f(x))$ and $\xi'(z) = e^T f'(z)$. So for any vectors x, y, there exists $z = \lambda x + (1-\lambda)y$ such that

$$e^T f(x) - e^T f(y) = e^T f'(z)(y - x),$$

which is exactly what needs to be proved. □

Lemma A.3. *Let $f : R^a \to R^b$ be any function. For any $x, y \in R^a$, there exists $\lambda \in (0,1)$ and $z = \lambda x + (1-\lambda)y$ such that*

$$||f(y) - f(x)|| = ||f'(z)||\ ||y - x||.$$

Proof. This is a direct result from Lemma A.2 by taking the l_2-norm on both sides and cancelling $||e||$. □

Lemma A.4. *The Projected Neural TD Learning with a mean-path update can be rewritten as*

$$\theta_{t+1} = \text{Proj}(\theta_t + \alpha_t(\bar{g}_1(\theta_t) + \bar{g}_2(\theta_t) + \bar{g}_3(\theta_t)))$$

where $\bar{g}_1(\theta_t), \cdots, \bar{g}_5(\theta_t)$ are defined as follows:

$$\bar{g}_1(\theta_t) = \nabla_\theta V(\theta_1^{\text{mid}})^T D(\gamma P - I)(V(\theta_t) - V(\hat{\theta}_*)), \tag{12}$$

$$\bar{g}_2(\theta_t) = (\nabla_\theta V(\theta_t) - \nabla_\theta V(\theta_1^{\text{mid}}))^T D(\gamma P - I)(V(\theta_t) - V(\hat{\theta}_*)), \tag{13}$$

$$\bar{g}_3(\theta_t) = \nabla_\theta V(\theta_t)^T D(\gamma P - I)(\hat{V}^* - V^*), \tag{14}$$

where $\lambda \in [0,1]$ is a scalar and $\theta_1^{\text{mid}} = \lambda\theta_t + (1-\lambda)\hat{\theta}_*$ is a vector such that

$$(\theta_t - \hat{\theta}_*)^T \nabla_\theta V(\theta_1^{\text{mid}})^T D(\gamma P - I)(V(\theta_t) - V(\hat{\theta}_*)) = (V(\theta_t) - V(\hat{\theta}_*))^T D(\gamma P - I)(V(\theta_t) - V(\hat{\theta}_*)). \tag{15}$$

Proof. By Eq.(8),

$$\begin{aligned}\bar{g}(\theta_t) &= \nabla_\theta V(\theta_t)^T D(\gamma P - I)(V(\theta_t) - V(\theta_*)) \\ &= \nabla_\theta V(\theta_t)^T D(\gamma P - I)(V(\theta_t) - V(\hat{\theta}_*)) + \nabla_\theta V(\theta_t)^T D(\gamma P - I)(\hat{V}^* - V^*).\end{aligned}$$

Now let $D(\gamma P - I)(V(\theta_t) - V(\hat{\theta}_*))$ be the vector e in Lemma A.2. There exists a scalar $\lambda \in (0,1)$ and a vector $\theta_1^{\text{mid}} = \lambda\theta_t + (1-\lambda)\hat{\theta}_*$ such that

$$(\theta_t - \hat{\theta}_*)^T \nabla_\theta V(\theta_1^{\text{mid}})^T D(\gamma P - I)(V(\theta_t) - V(\hat{\theta}_*)) = (V(\theta_t) - V(\hat{\theta}_*))^T D(\gamma P - I)(V(\theta_t) - V(\hat{\theta}_*)).$$

So the term $\nabla_\theta V(\theta_t)^T D(\gamma P - I)(V(\theta_t) - V(\hat{\theta}_*))$ can also be divided into two parts,

$$\begin{aligned}&\nabla_\theta V(\theta_t)^T D(\gamma P - I)(V(\theta_t) - V(\hat{\theta}_*)) \\ =&\nabla_\theta V(\theta_1^{\text{mid}})^T D(\gamma P - I)(V(\theta_t) - V(\hat{\theta}_*)) + (\nabla_\theta V(\theta_t) - \nabla_\theta V(\theta_1^{\text{mid}}))^T D(\gamma P - I)(V(\theta_t) - V(\hat{\theta}_*)).\end{aligned}$$

□

Lemma A.5. *Let x, y be two vectors and $x(i), y(i)$ denote their i'th entry, respectively. Let $\bar{x}, \bar{y}$ be two scalars such that $|x(i)| \le \bar{x}$ and $|y(i)| \le \bar{y}$ hold for all i. The following results hold:*

$$x^T D y = y^T D x \le \bar{x}\bar{y},$$
$$x^T D P y \le \bar{x}\bar{y},$$
$$y^T D P x \le \bar{x}\bar{y}.$$

Proof. We expand $x^T D y$, $x^T D P y$ and $y^T D P x$ as follows:

$$x^T D y = y^T D x = \sum_i \mu(s_i) x(i) y(i) \le \sum_i \mu(s_i)\bar{x}\bar{y} \le \bar{x}\bar{y},$$

$$x^T D P y = \sum_i \mu(s_i) x(i) \sum_j P(s_j|s_i) y(j) \le \bar{x}\bar{y} \sum_i \mu(s_i) \sum_j P(s_j|s_i) \le \bar{x}\bar{y},$$

$$y^T D P x = \sum_i \mu(s_i) y(i) \sum_j P(s_j|s_i) x(j) \le \bar{x}\bar{y} \sum_j \sum_i \mu(s_i) P(s_j|s_i) \le \bar{x}\bar{y}.$$

□

Lemma A.6. *For all $k \in \{1, 2, \ldots, K\}$,*

$$||\theta^{(k)}|| \leq O(\sqrt{m}).$$

Proof.

$$\begin{aligned}
||\theta^{(k)}|| &\leq ||\theta^{(k)} - \theta_0^{(k)}|| + ||\theta_0^{(k)}|| \\
&\leq ||\theta - \theta_0|| + ||\theta_0^{(k)}|| \\
&\leq \omega + ||\theta_0^{(k)}|| \\
&\leq O(\sqrt{m}),
\end{aligned}$$

where the last inequality uses Assumption 3.1. □

Lemma A.7. *For all $k \in \{1, 2, \ldots, K\}$,*

$$||x^{(k)}|| \leq O(\sqrt{m}).$$

Proof. From Assumption 2.3, $||x^{(0)}|| \leq 1$. By Lemma A.6,

$$\begin{aligned}
||x^{(1)}||^2 &= ||\frac{1}{\sqrt{m}}\sigma(\theta^{(1)}x^{(0)})||^2 \\
&\leq \frac{1}{m}l^2||\theta^{(1)}||^2||x^{(0)}||^2 + m|\sigma(0)|^2 \\
&\leq O(m).
\end{aligned}$$

By induction, suppose $||x^{(k)}||^2 \leq O(m)$. By Lemma A.6,

$$\begin{aligned}
||x^{(k+1)}||^2 &= ||\frac{1}{\sqrt{m}}\sigma(\theta^{(k+1)}x^{(k)})||^2 \\
&\leq \frac{1}{m}l^2||\theta^{(k+1)}||^2||x^{(k)}||^2 + m|\sigma(0)|^2 \\
&\leq O(m).
\end{aligned}$$

□

Lemma A.8. *For all $k \in \{1, 2, \ldots, K\}$,*

$$||\nabla_{x^{(k-1)}}x^{(k)}|| \leq O(1).$$

Proof.

$$\nabla_{x^{(k-1)}}x^{(k)}(i,j) = \frac{1}{\sqrt{m}}\sigma'(\sum_j \theta^{(k)}(i,j)x^{(k-1)}(j))\theta^{(k)}(i,j),$$

which implies

$$\begin{aligned}
||\nabla_{x^{(k-1)}}x^{(k)}||^2 &= \sup_{||v||=1}\sum_{i=1}^{m}(\sum_j \nabla_{x^{(k-1)}}x^{(k)}(i,j)v_j)^2 \\
&= \sup_{||v||=1}\frac{1}{m}||\Sigma'\theta^{(k)}v||^2 \\
&\leq \frac{1}{m}||\Sigma'||^2||\theta^{(k)}||^2 \\
&\leq O(1).
\end{aligned}$$

where Σ' is a diagonal matrix with $\Sigma'(i,i) = \sigma'(\sum_j \theta^{(k)}(i,j)x^{(k-1)}(j))$. The last inequality holds because of Lemma A.6. □

Lemma A.9. *For all $k \in \{1, 2, \ldots, K\}$,*

$$||\nabla_{\theta^{(k)}} x^{(k)}|| \leq O(1)$$

Here, $\nabla_{\theta^{(k)}} x^{(k)}$ is defined to be a matrix whose $(i, (j-1)m+h)$'th entry $\nabla_{\theta^{(k)}} x^{(k)}(i, j, h)$ is given by

$$\nabla_{\theta^{(k)}} x^{(k)}(i, j, h) = \frac{\partial x^{(k)}(i)}{\partial \theta^{(k)}(j, h)}.$$

Proof.

$$\nabla_{\theta^{(k)}} x^{(k)}(i, j, j') = \frac{1}{\sqrt{m}} 1\{i = j\} \sigma'(\sum_h \theta^{(k)}(i, h) x^{(k-1)}(h)) x^{(k-1)}(j'),$$

which implies

$$\begin{aligned}
||\nabla_{\theta^{(k)}} x^{(k)}||^2 &= \sup_{||V||_F=1} \sum_{i=1}^{m} (\nabla_{\theta^{(k)}} x^{(k)}(i, j, j') V_{j,j'})^2 \\
&= \sup_{||V||_F=1} \frac{1}{m} ||\Sigma' V x^{(k-1)}||^2 \\
&\leq \frac{1}{m} ||\Sigma'||^2 ||x^{(k-1)}||^2 \\
&\leq O(1).
\end{aligned}$$

The last inequality holds because of Lemma A.7. □

Lemma A.10. *For all $s \in S$ and θ,*

$$||\nabla_\theta V(s, \theta)|| \leq O(1).$$

with respect to m.

Proof. Since each entry of b satisfies $|b_r| \leq 1$, it is easy to see that

$$||\nabla_{x^{(K)}} V(s, \theta)|| = \frac{1}{\sqrt{m}} ||b|| \leq 1.$$

By Lemma A.8 and Lemma A.9,

$$||\nabla_{\theta^{(k)}} V(s, \theta)|| = ||\nabla_{x^{(K)}} V(s, \theta) \nabla_{x^{(K-1)}} x^{(K)} \cdots \nabla_{x^{(k)}} x^{(k+1)} \nabla_{\theta^{(k)}} x^{(k)}|| \leq O(1).$$

It follows:

$$||\nabla_\theta V(s, \theta)||^2 = \sup_{||V||_F=1} \sum_{k=1}^{K} (\nabla_{\theta^{(k)}} V(s, \theta) V_k)^2 \leq O(1).$$

□

Lemma A.11. *The Hessian matrix, which is $\nabla_\theta^2 V(s, \theta)$, has a norm of $O(m^{-0.5})$. In other words,*

$$||\nabla_\theta^2 V(s, \theta)|| \leq O(m^{-0.5}).$$

This is a direct result from Liu et al. (2020).

A.2 Proof for The Main Result

Proof. Consider Projected Neural TD Learning, it is easy to see

$$\begin{aligned}
||\theta_{t+1} - \hat{\theta}_*||^2 &= ||\mathbf{Proj}(\theta_t + \alpha_t g(\theta_t)) - \hat{\theta}_*||^2 \\
&\leq ||\theta_t - \hat{\theta}_* + \alpha_t g(\theta_t)||^2 \\
&= ||\theta_t - \hat{\theta}_*||^2 + 2\alpha_t (\theta_t - \hat{\theta}_*)^T g(\theta_t) + \alpha_t^2 ||g(\theta_t)||^2 \\
&= ||\theta_t - \hat{\theta}_*||^2 + 2\alpha_t (\theta_t - \hat{\theta}_*)^T \bar{g}(\theta_t) + \alpha_t^2 ||g(\theta_t)||^2 + 2\alpha_t (\theta_t - \hat{\theta}_*)^T (g(\theta_t) - \bar{g}(\theta_t)).
\end{aligned} \tag{16}$$

First, we consider $2\alpha_t(\theta_t - \hat{\theta}_*)^T\bar{g}(\theta_t)$. Lemma A.4 allows us to divide it into several parts and bound them respectively. By Eq.(12), Eq.(15) and letting $f = V(\theta_t) - V(\hat{\theta}_*)$ in Lemma A.1,

$$(\theta_t - \hat{\theta}_*)^T\bar{g}_1(\theta_t) = -\mathcal{N}(V(\theta_t) - V(\hat{\theta}_*)).$$

By Lemma A.11 and Lemma A.3,

$$||\nabla_\theta V(s,\theta_t) - \nabla_\theta V(s,\theta_1^{\text{mid}})|| \leq (1-\lambda)O(m^{-0.5})||\theta_t - \hat{\theta}_*|| \leq O(m^{-0.5}).$$

By Lemma A.10 and Lemma A.3,

$$||V(s,\theta_t) - V(s,\hat{\theta}_*)|| = O(1)||\theta_t - \hat{\theta}_*|| \leq O(1).$$

By Lemma A.5 and Eq.(13),

$$(\theta_t - \hat{\theta}_*)^T\bar{g}_2(\theta_t) \leq O(m^{-0.5}).$$

By Lemma A.5 and Eq.(14),

$$(\theta_t - \hat{\theta}_*)^T\bar{g}_3(\theta_t) \leq O(\epsilon).$$

Combining the above facts,

$$2\alpha_t(\theta_t - \hat{\theta}_*)^T\bar{g}(\theta_t) \leq -2\alpha_t\mathcal{N}(V(\theta_t) - V(\hat{\theta}_*)) + O(\alpha_t(m^{-0.5} + \epsilon))$$

which is the bound of first part.

Second, we consider $\alpha_t^2||g(\theta_t)||^2$. For simplicity, define $\hat{f}(s) = V(s,\hat{\theta}_*) - V(s,\theta_t)$. By Lemma A.10,

$$||\hat{f}(s)||^2 \leq O(1).$$

Using the above fact and Jensen's inequality,

$$||\gamma\sum_{s'}P(s'|s)\hat{f}(s')||^2 \leq \gamma^2\sum_{s'}P(s'|s)||\hat{f}(s)||^2 \leq O(1).$$

On the other hand, Eq.(1) can be rewritten as

$$V^*(s') = r(s) + \gamma\sum_{s''}P(s''|s)V^*(s''), \tag{17}$$

where $g(\theta_t)$ is actually a random variable and implicitly relied on the state s. This is easy to see from the definition of $g(\theta_t)$, which is

$$g(\theta_t) = \nabla_\theta V(s,\theta_t)(r(s) + \gamma V(s',\theta_t) - V(s,\theta_t)).$$

Hence, $g(\theta_t)$ can be rewritten as

$$\begin{aligned}
g(\theta_t) &= \nabla_\theta V(s,\theta_t)(V^*(s) - V(s,\theta_t) + \gamma\sum_{s''}P(s''|s)(V(s',\theta_t) - V^*(s''))) \\
&= \nabla_\theta V(s,\theta_t)(\hat{f}(s) + V^*(s) - V(s,\hat{\theta}_*) + \gamma\sum_{s''}P(s''|s)(-\hat{f}(s') \\
&\quad + V(s',\hat{\theta}_*) - V(s'',\hat{\theta}_*) + V(s'',\hat{\theta}_*) - V^*(s''))).
\end{aligned}$$

By Lemma A.10, $||g(\theta_t)||^2$ can be bounded as

$$\begin{aligned}
||g(\theta_t)||^2 &\leq 5||\nabla_\theta V(s,\theta_t)||^2(||\hat{f}(s)||^2 + ||\gamma\sum_{s''}P(s''|s)\hat{f}(s'')||^2 + \epsilon^2 + \gamma^2\epsilon^2 \\
&\quad + ||\gamma\sum_{s''}P(s''|s)(V(s',\hat{\theta}_*) - V(s'',\hat{\theta}_*))||^2) \\
&\leq O(1+\epsilon^2) + \gamma^2||V(s',\hat{\theta}_*) - V(s,\hat{\theta}_*)||^2 \\
&\leq O(1+\epsilon+\epsilon^2)
\end{aligned} \tag{18}$$

where the last inequality is because of Eq.(10). Further,

$$\alpha_t^2||g(\theta_t)||^2 \leq O(\alpha_t^2(1+\epsilon+\epsilon^2))$$

Above facts are all we need to establish the result when each s_t are sample i.i.d. from μ. In summary, $2\alpha_t(\theta_t-\hat{\theta}_*)^T\bar{g}(\theta_t)+\alpha_t^2||g(\theta_t)||^2 \leq -2\alpha_t\mathcal{N}(V(\theta_t)-V(\hat{\theta}_*))+O(\alpha_t(m^{-0.5}+\epsilon))+O(\alpha_t^2(1+\epsilon+\epsilon^2))$. Given θ_t, taking expectation on both sides of Eq.(16):

$$\begin{aligned}
&\mathbb{E}[||\theta_{t+1}-\hat{\theta}_t||^2] \\
\leq&||\theta_t-\hat{\theta}_*||^2+2\alpha_t(\theta_t-\hat{\theta}_*)\bar{g}(\theta_t)+\mathbb{E}[\alpha_t^2||g(\theta_t)||^2]+2\alpha_t(\theta_t-\hat{\theta}_*)\mathbb{E}[g(\theta_t)-\bar{g}(\theta_t)] \\
\leq&||\theta_t-\hat{\theta}_*||^2-2\alpha_t\mathcal{N}(V(\theta_t)-V(\hat{\theta}_*))+O(\alpha_t(m^{-0.5}+\epsilon))+O(\alpha_t^2(1+\epsilon+\epsilon^2)),
\end{aligned}$$

where the second inequality uses the fact that given s_t are sampled i.i.d. from μ, $\mathbb{E}[g(\theta_t)-\bar{g}(\theta_t)]=0$. If $\alpha_t=\alpha$ are constant over t,

$$\frac{\mathbb{E}[||\theta_{t+1}-\hat{\theta}_*||^2]}{2\alpha}-\frac{\mathbb{E}[||\theta_t-\hat{\theta}_*||^2]}{2\alpha} \leq -\mathbb{E}[\mathcal{N}(V(\theta_t)-V(\hat{\theta}_*))]+O(m^{-0.5}+\epsilon)+O(\alpha_t(1+\epsilon+\epsilon^2)).$$

Telescoping a sum from 1 to T and dividing both sides by T, we establish the result in i.i.d. case.

To continue with the non i.i.d. case, we need to consider $2\alpha_t(\theta_t-\hat{\theta}_*)^T(g(\theta_t)-\bar{g}(\theta_t))$. Our first step is to divide it into two terms:

$$2\alpha_t(\theta_t-\hat{\theta}_*)(g(\theta_t)-\bar{g}(\theta_t))=2\alpha_t(\theta_t-\theta_{t-\tau_{\text{mix}}})(g(\theta_t)-\bar{g}(\theta_t))+2\alpha_t(\theta_{t-\tau_{\text{mix}}}-\hat{\theta}_*)(g(\theta_t)-\bar{g}(\theta_t)), \tag{19}$$

where τ_{mix} is defined in Eq.(3). We will bound each of these two terms separately.

To bound $2\alpha_t(\theta_t-\theta_{t-\tau_{\text{mix}}})(g(\theta_t)-\bar{g}(\theta_t))$, notice that Eq.(18) implies $||g(\theta_t)|| \leq O(1+\epsilon)$. Further, the same upper bound holds for $\bar{g}(\theta_t)$ since $\bar{g}(\theta_t)=\mathbb{E}[g(\theta_t)]$. Finally, as a consequence of Eq.(6), we have

$$||\theta_t-\theta_{t-\tau_{\text{mix}}}|| \leq (O(1+\epsilon))\alpha_t\tau_{\text{mix}}.$$

These observations imply that

$$2\alpha_t(\theta_t-\theta_{t-\tau_{\text{mix}}})(g(\theta_t)-\bar{g}(\theta_t)) \leq 4(O(1)+O(\epsilon))\alpha_t^2\tau_{\text{mix}}.$$

To bound $2\alpha_t(\theta_{t-\tau_{\text{mix}}}-\hat{\theta}_*)(g(\theta_t)-\bar{g}(\theta_t))$, observe that, conditioned on $\theta_{t-\tau_{\text{mix}}}$, the quantity $\theta_{t-\tau_{\text{mix}}}-\hat{\theta}_*$ is deterministic, and $\mathbb{E}[g(\theta_t)-\bar{g}(\theta_t)|\theta_{t-\tau\text{mix}}]$ can be bounded by

$$\begin{aligned}
\mathbb{E}[g(\theta_t)-\bar{g}(\theta_t)|\theta_{t-\tau\text{mix}}] &\leq (O(1+\epsilon))\max_s ||P_{s,:}^{\tau_{\text{mix}}}-\mu||_{\text{TV}} \\
&=(O(1+\epsilon))\epsilon_{\text{mix}}.
\end{aligned}$$

The first inequality follows because, conditional on $s_{t-\tau_{\text{mix}}}$, the random variable s_t has the distribution of one row of $P^{\tau_{\text{mix}}}$ (and we have already mentioned that $||g(\theta_t)-\bar{g}(\theta_t)||=2O(1+\epsilon)$). Thus the second term in Eq.(19) can be bounded as

$$\begin{aligned}
\mathbb{E}[2\alpha_t(\theta_{t-\tau_{\text{mix}}}-\hat{\theta}_*)(g(\theta_t)-\bar{g}(\theta_t))] &= \mathbb{E}[\mathbb{E}[2\alpha_t(\theta_{t-\tau_{\text{mix}}}-\hat{\theta}_*)(g(\theta_t)-\bar{g}(\theta_t))|\theta_{t-\tau_{\text{mix}}}]] \\
&=4(O(1+\epsilon))\alpha_t\omega\epsilon_{\text{mix}}.
\end{aligned}$$

Thus coming back to Eq.(19), we obtain that

$$\mathbb{E}[2\alpha_t(\theta_t-\hat{\theta}_*)(g(\theta_t)-\bar{g}(\theta_t))] \leq 4(O(1+\epsilon))\alpha_t^2\tau_{\text{mix}}+4(O(1+\epsilon))\alpha_t\omega\epsilon_{\text{mix}}.$$

Now let $\epsilon_{\text{mix}}=\alpha_t$, by the definition of τ_{mix}, $\tau_{\text{mix}}=\frac{\log\frac{\alpha_t}{C}}{\log\beta}$. Using the fact that $\log x \leq x-1, \forall x>0$, we derive

$$\begin{aligned}
\frac{\log\frac{\alpha_t}{C}}{\log\beta} &\leq \frac{\log\frac{\alpha_t}{C}}{\beta-1} \\
&=\frac{\log\frac{C}{\alpha_t}}{1-\beta},
\end{aligned}$$

where the first inequality is because $\log\beta \leq \beta-1<0$ and $\log\frac{\alpha_t}{C} \leq 0$. The bound can be rewritten as

$$\mathbb{E}[2\alpha_t(\theta_t-\hat{\theta}_*)(g(\theta_t)-\bar{g}(\theta_t))] \leq O((1+\epsilon)\alpha_t^2\frac{\log\frac{C}{\alpha_t}}{1-\beta}).$$

Now we are ready to consider Eq.(16). Taking expectation on both sides,

$$\begin{aligned}&\mathbb{E}[2\alpha_t(\theta_t-\hat{\theta}_*)^T g(\theta_t)+\alpha_t^2||g(\theta_t)||^2]\\ \leq&-2\alpha_t\mathcal{N}(V(\theta_t)-V(\hat{\theta}_*))+O(\alpha_t(m^{-0.5}+\epsilon))+O(\alpha_t^2(1+\epsilon+\epsilon^2))+O((1+\epsilon)\alpha_t^2\frac{\log\frac{C}{\alpha_t}}{1-\beta}).\end{aligned}$$

If $\alpha_t=\alpha$ are constant over t,

$$\begin{aligned}\frac{\mathbb{E}[||\theta_{t+1}-\hat{\theta}_*||^2]}{2\alpha}-\frac{\mathbb{E}[||\theta_t-\hat{\theta}_*||^2]}{2\alpha}\leq&-\mathbb{E}[\mathcal{N}(V(\theta_t)-V(\hat{\theta}_*))]+O(m^{-0.5}+\epsilon)\\ &+O(\alpha(1+\epsilon+\epsilon^2))+O((1+\epsilon)\alpha^2\frac{\log\frac{C}{\alpha}}{1-\beta}).\end{aligned}$$

Telescoping a sum from 1 to T and dividing both sides by T, we establish the result in non i.i.d. case. □