# Understanding and Leveraging Overparameterization in Recursive Value Estimation

**Anonymous authors**
Paper under double-blind review

## Abstract

The theory of function approximation in reinforcement learning (RL) typically considers low capacity representations that incur a tradeoff between approximation error, stability and generalization. Current deep architectures, however, operate in an overparameterized regime where approximation error is not necessarily a bottleneck. To better understand the utility of deep models in RL we present an analysis of recursive value estimation using *overparameterized* linear representations that provides useful, transferable findings. First, we show that classical updates such as temporal difference (TD) learning or fitted-value-iteration (FVI) converge to *different* fixed points than residual minimization (RM) in the overparameterized linear case. We then develop a unified interpretation of overparameterized linear value estimation as minimizing the Euclidean norm of the weights subject to alternative constraints. A practical consequence is that RM can be modified by a simple alteration of the backup targets to obtain the same fixed points as FVI and TD (when they converge), while universally ensuring stability. Further, we provide an analysis of the generalization error of these methods, demonstrating per iterate bounds on the value prediction error of FVI, and fixed point bounds for TD and RM. Given this understanding, we then develop new algorithmic tools for improving recursive value estimation with deep models. In particular, we extract two regularizers that penalize out-of-span top-layer weights and co-linearity in top-layer features respectively. Empirically we find that these regularizers dramatically improve the stability of TD and FVI, while allowing RM to match and even sometimes surpass their generalization performance with assured stability.

## 1 Introduction

Model-free value estimation remains a core method of reinforcement learning (RL), lying at the heart of some of the most prominent achievements in this area (Mnih et al., 2015; Bellemare et al., 2020). Such success appears paradoxical however, given that value estimation is subject to the *deadly triad*: any value update that combines off-policy estimation with Bellman-bootstrapping and function approximation diverges in the worst case (Sutton and Barto, 2018). Without additional assumptions it is impossible to ensure the viability of iterative value estimation schemes, yet this remains a dominant method in RL—its popularity supported by empirical success in many applications. Such a sizable gap between theory and practice reflects limited understanding of such methods, how they behave in practice, and what accounts for their empirical success (van Hasselt et al., 2018; Achiam et al., 2019).

Decomposing the deadly triad indicates that off-policy estimation and bootstrapping are difficult to forego: off-policy estimation is supported by the empirical effectiveness of action value maximization and replay buffers, while Bellman-bootstrapping provides significant advantages over Monte Carlo estimation (Sutton, 1988). On the other hand, our understanding of the third factor, the relationship between function representation and generalization, has evolved dramatically in recent years. Although it was once thought that *restrictive* function approximation—representations that lack capacity to fit all data constraints—might be essential for generalization, we now know that this view is oversimplified (Belkin et al., 2019). The empirical success of deep learning (Krizhevsky et al., 2012), extremely large models (Brown et al., 2020) and associated theoretical advances (Jacot et al., 2018) have made it clear that gradient-based training of overparameterized models embodies implicit biases that encourage generalization even after all data constraints are fit exactly. This success suggests a new opportunity for breaking the deadly triad: by leveraging overparameterized value representations one can avoid some of the most difficult tradeoffs in value-based RL (Lu et al., 2018).

The use of overparameterized deep models in value-based RL, however, still exhibits mysteries in stability and performance. Although one might expect larger capacity models to improve the stability of Bellman-bootstrapping, in fact the opposite appears to occur (van Hasselt et al., 2018). Our own empirical experience indicates that classical value estimation with deep models always diverges eventually in non-toy problems. It has also been shown that value updating leads to premature rank-collapse in deep models (Kumar et al., 2021), coinciding with instability and degrading generalization. In practice, some form of *early-stopping* is usually necessary to obtaining successful results, a fact that is not often emphasized in the literature (Agarwal et al., 2021). Meanwhile, there is a long history of convergent methods being proposed in the RL literature—starting from residual gradient (Baird, 1995), to gradient-TD (Sutton et al., 2008; Maei et al., 2009), prox gradient TD (Liu et al., 2015; 2016), and emphatic TD (Yu, 2015; Sutton et al., 2016)—yet none of these has demonstrated sufficient generalization quality to supplant unstable methods. The current state of development leaves an awkward tradeoff between stability and generalization. A stable recursive value estimation method that ensures generalization quality with overparametrization remains elusive.

In this paper we investigate whether overparameterized value representations might allow the stability-generalization tradeoff to be better managed, enabling stable estimation methods that break the deadly triad and generalize well. To this end, we first consider policy evaluation with *overparameterized linear* value representations, a simplified setting that still imposes the deadly triad (Zhang et al., 2021). Here we first show that alternative updates, such as temporal differencing (TD), fitted value iteration (FVI) and residual minimization (RM) converge to *different* fixed points in the overparameterized case (when they converge), even though these updates share a common fixed point when the approximation error is zero and there are no extra degrees of freedom (Dann et al., 2014). That is, these algorithms embody certain implicit biases that *only* become distinguishable in the overparameterized case. From this result, we observe that the fixed points lie in different bases, which we use to develop a unified view of iterative value estimation as minimizing the Euclidean norm of the weights subject to alternative constraint sets. This unification allows us to formulate alternative updates that share common fixed points with TD and FVI but guarantee stability without requiring regularization or prox constraints (Zhang et al., 2021). Next, we analyze the generalization performance of these algorithms and provide a per-iterate bound on the value estimation error of FVI, and fixed point bounds on the value estimation error of TD. From these results, we identify two novel regularizers, one that closes the gap between RM and TD and another that quantifies the effect of the feature representation on the generalization bound. We deploy these regularizers in a realistic study of deep model training for optimal value estimation and observe systematic stability and generalization improvements. We also observe that the performance gap between RM and TD/FVI can be closed and in some cases eliminated.

## 2 Related work

Value estimation has a lengthy history throughout RL research. Our main focus is on off-policy value estimation with parametric function representations and iterative (i.e., gradient based) updates. We do not consider exploration nor full planning problems (i.e., approximately solving an entire Markov decision process (MDP)) in the theoretical development, but instead focus on offline value estimation; however, we do apply the findings to policy improvement experiments in the empirical investigation.

Dann et al. (2014) provide a comprehensive survey of value estimation with parametric function representations. Significant attention has been focused on *under*parameterized representations where backed up values are not necessarily expressible in the function class, however we focus on the overparameterized case where any backed up values can be assumed to be exactly representable with respect to finite data. This change fundamentally alters the conclusions one can draw about algorithm behavior, as we see below. One of the key consequences is that classical distinctions (Scherrer, 2010; Dann et al., 2014) between objectives—e.g., mean squared Bellman error (MSBE), mean squared *projected* Bellman error (MSPBE), mean squared temporal difference error (MSTDE), and the norm of the expected TD update (NEU)—all collapse when the Bellman errors can all be driven to zero. Despite this collapse, we find that algorithms targetting the different objectives—TD and LSTD for MSPBE (Sutton, 1988; Bradtke and Barto, 1996) and RM without double sampling (DS) for MSTDE (Maei et al., 2009; Dann et al., 2014)—converge to different fixed points given overparameterization, even when they ultimately satisfy the same set of temporal consistency constraints.

It is well known that classical value updates can diverge given off-policy data and parametric function representations (Baird, 1995; Tsitsiklis and Van Roy, 1996; 1997). The stability of these methods has therefore been studied extensively with many mitigations proposed, including restricting the function representation (Gordon, 1995; Szepesvári and Smart, 2004) or adjusting the representation to ensure contraction (Kolter, 2011; Ghosh and Bellemare, 2020; Wang et al., 2021b), or modifying the updates to achieve convergent variations, such as LSTD (Bradtke and Barto, 1996; Yu, 2010), FVI (Ernst et al., 2005; Munos and Szepesvári, 2005; Szepesvári and Munos, 2008; Lizotte, 2011) or the introduction of target networks (Mnih et al., 2015; Lillicrap et al., 2016; Zhang et al., 2021; Carvalho et al., 2020). Others have considered modified the updates to combat various statistical inefficiencies (van Hasselt, 2010; Weng et al., 2020; Konidaris et al., 2011). Another long running trend has been to consider two time-scale algorithms and analyses, reflected in gradient-TD methods (Sutton et al., 2008; Maei et al., 2009), prox gradient TD (Liu et al., 2015; 2016), primal-dual TD (Dai et al., 2017; Du et al., 2017), and emphatic TD (Yu, 2015; Sutton et al., 2016). Beyond mere convergence, however, we discover a greater diversity in fixed points among algorithms in the overparameterized case, which play a critical but previously unacknowledged role in generalization quality.

The fact that minimizing MSPBE via TD methods still dominates practice appears surprising given the theoretical superiority of other objectives. It has been argued, for example, that direct policy gradient methods (Sutton et al., 1999) dominate minimizing Bellman error objectives (Geist et al., 2017). Even among Bellman based approaches, it is known that MSBE can upper bound the value estimation error (MSE) whereas MSPBE cannot (Kolter, 2011; Dann et al., 2014), yet MSPBE minimization (via TD based methods) empirically dominates minimizing MSBE (via residual methods). This dominance has been thought to be due to the double sampling bias of residual methods (Baird, 1995; Dann et al., 2014), but we uncover a more interesting finding that their fixed points lie in different bases in the overparameterized setting, and that reducing this difference closes the performance gap.

We analyze the convergence of classical updates given offline data and provide associated generalization bounds, with the primary goal of understanding the discrepancy between previous theory and the empirical success of TD/FVI versus RM. Although this theory sheds new light in exploitable ways, it cannot overcome theoretical limits on offline value estimation, such as lower bounds on worst case error that are exponential in horizon length (Wang et al., 2021a;b; Zanette, 2021; Xiao et al., 2021). We analyze the convergence of the expected updates, extendible to the stochastic case using known techniques (Yu, 2010; Bhandari et al., 2018; Dalal et al., 2018; Prashanth et al., 2021; Patil et al., 2021). We expand the coverage of these earlier works by including alternative updates and focusing on the overparameterized case, uncovering previously unobserved differences in the fixed points.

There is a growing body of work on linear value estimation and planning that leverages the insight of (Parr et al., 2008; Taylor and Parr, 2009) that linear value estimation is equivalent to linear model approximation. A number of works have strived to obtain provably efficient algorithms for approximating the optimal policy values in this setting, but these generally rely on exploration or strong assumptions about data coverage (Song et al., 2016; Yang and Wang, 2019; Duan et al., 2020; Agarwal et al., 2020; Jin et al., 2020; Yang et al., 2020; Hao et al., 2021) that we do not make. Instead we study linear value estimation to gain insight, but rather than focus on linear planning we leverage the findings to improve the empirical performance of value estimation with deep models.

## 3 Preliminaries

**Notation** We let $\mathbb{R}$ denote the set of real numbers, $\boldsymbol{I}_n$ an $n \times n$ identity matrix, and $\mathbb{I}$ the indicator function. For a finite set $\mathcal{X}$, we use $\Delta(\mathcal{X})$ to denote the set of probability distributions over $\mathcal{X}$. For a vector $\mu$ we let $|\text{supp}(\mu)|$ denote the size of the support of $\mu$ (i.e., the number of nonzero entries in $\mu$). For a matrix $\boldsymbol{A} \in \mathbb{R}^{n\times m}$, we let $\boldsymbol{A}^\dagger$ be the Moore-Penrose pseudoinverse of $\boldsymbol{A}$, $\|\boldsymbol{A}\|$ be its spectral norm, $\lambda_{\max}(\boldsymbol{A})$ and $\lambda_{\min}(\boldsymbol{A})$ be its maximum and minimum *non-zero* eigenvalues. We also use $\Pi_{\boldsymbol{A}} = \boldsymbol{A}^\dagger \boldsymbol{A}$ to denote the projection matrix to the row space of $\boldsymbol{A}$. For a vector $x \in \mathbb{R}^d$, we let $\|x\|$ be its $l_2$ norm and $\|x\|_{\boldsymbol{A}} = \sqrt{x^\top \boldsymbol{A} x}$ be the associated norm for a positive definite matrix $\boldsymbol{A}$. We also use $\text{diag}(x) \in \mathbb{R}^{d\times d}$ to denote a diagonal matrix whose diagonal elements are $x$.

**Markov reward processes** We consider the problem of predicting the value of a given stationary policy in a Markov Decision Process (MDP). For a stationary policy, this problem can naturally formulated in terms of a Markov reward process $M = \{\mathcal{S}, P, r, \gamma\}$, such that $\mathcal{S}$ is a finite set of states, $r : \mathcal{S} \to \mathbb{R}$ and $P : \mathcal{S} \to \Delta(\mathcal{S})$ are the reward and transition functions respectively, and $\gamma \in [0, 1)$ is the discount factor. For a given state $s \in \mathcal{S}$, the function $r(s)$ gives the immediate reward incurred at

$s$, while $P(\cdot|s)$ gives the next-state transition probability of $s$. The value function specifies the future discounted total reward obtained from each state, which is defined as

$$v(s) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t) \Big| s_0 = s\right]. \tag{1}$$

To simplify the presentation we identify functions as vectors to allow vector-space operations. In particular, the reward function $r$ is identified as a vector $r \in \mathbb{R}^{|\mathcal{S}|}$, and the transition $P$ is identified as an $|\mathcal{S}| \times |\mathcal{S}|$ transition matrix, where the $s$-th row $P_s$ specifies the transition probability $P(\cdot|s)$ of state $s$. These definitions allow the value function to be expressed using Bellman's equation

$$v = r + \gamma P v\,. \tag{2}$$

**Linear Function Approximation** It is usually not possible to consider tabular value representations in practice, since the state set is usually combinatorial or infinite. In our theoretical development we focus on linear function approximations, where $v$ is approximated by a linear combination of features describing states; i.e., $v(s) \approx \phi(s)^\top \theta$, where $\theta \in \mathbb{R}^d$ is a parameter vector and $\phi : \mathcal{S} \to \mathbb{R}^d$ maps a given state $s \in \mathcal{S}$ to a $d$-dimensional feature vector $\phi(s) \in \mathbb{R}^d$. We let $\Phi \in \mathbb{R}^{|\mathcal{S}| \times d}$ denote the feature matrix, with the $s$-th row corresponding to the feature vector $\phi(s)$, so that the value approximation can be written as $v \approx \Phi\theta$. We assume $\|\phi(s)\| \le 1$ for any $s \in \mathcal{S}$, and for simplicity we also assume that there is no redundant or irrelevant features in the feature map; that is, $\Phi$ is full rank.

### 3.1 Batch Value Estimation

We consider *batch mode* ("offline") estimation of the value function. Let $\mu \in \Delta(\mathcal{S})$ be an arbitrary probability distribution over states and $\boldsymbol{D}_\mu = \text{diag}(\mu)$. The data set consists of $\{s_i, r_i, s_i'\}_{i=1}^n$ transition tuples, which are generated by $s \sim \mu, r_i = r(s_i), s_i' \sim P(\cdot|s_i)$. Let $n(s) = \sum_{i=1}^n \mathbb{I}(s_i = s)$ be the number of counts of state $s$. We define the *empirical data distribution matrix* $\boldsymbol{D} = \text{diag}(\hat{\mu})$, where $\hat{\mu}(s) = n(s)/n$ is the empirical data distribution over states. The goal is to estimate the value function by finding a weight vector $\theta \in \mathbb{R}^d$ that minimizes the *value prediction error*,

$$\mathcal{E}(\theta) = \|\Phi\theta - v\|_{\boldsymbol{D}_\mu}^2 = \sum\nolimits_{s \in \mathcal{S}} \mu(s)(\phi(s)^\top\theta - v(s))^2\,. \tag{3}$$

Let $\hat{P}$ be the empirical transition matrix, where the $s$-th row represents the estimated transition of state $s$: if $n(s) > 0$, $\hat{P}_s(s') = \sum_{i=1}^n \mathbb{I}(s_i = s, s_i' = s')/n(s)$; if $n(s) = 0$, $\hat{P}_s(s') = 0$ for all $s' \in \mathcal{S}$. The empirical mean squared *Bellman error* on the batch data can be defined as

$$\text{MSBE}(\theta) = \tfrac{1}{2}\left\|r + \gamma\hat{P}\Phi\theta - \Phi\theta\right\|_{\boldsymbol{D}}^2. \tag{4}$$

**Over vs Underparameterized Features** In this paper we are particularly interested in the *overparameterized* regime $d > |\text{supp}(\hat{\mu})|$ where one can exactly satisfy the temporal consistencies on all transitions in the batch data set, achieving zero Bellman error. (Obviously this would also be possible if $d = |\text{supp}(\hat{\mu})|$ but the strictly overparameterized case is more interesting, as we will see below.) By contrast, in the underparameterized regime $d < |\text{supp}(\hat{\mu})|$, one can only expect to find an approximate solution that in general has nonzero Bellman error.

We consider three core algorithms in our analysis, covering major classical approaches.

**Residual Minimization (RM)** RM directly minimizes the empirical mean squared Bellman error Eq. (4) (MSBE) (Baird, 1995). The gradient update (Dann et al., 2014) can be expressed as

$$\theta_{t+1} = \theta_t - \eta(\Phi - \gamma\hat{P}\Phi)^\top \boldsymbol{D}\big(\Phi\theta_t - (r + \gamma\hat{P}\Phi\theta_t)\big)\,, \tag{5}$$

where $\theta_t$ is the estimated weight at step $t$, and $\eta$ is the learning rate. As a gradient descent method, the convergence of this update is robust, and applies to both linear and nonlinear function approximation.

**Temporal Difference (TD) Learning** The simplest variant of TD (Sutton, 1988), known as TD(0), also updates weights iteratively using transition data to approximate the value function. Let $\theta_t$ be the weight vector at step $t$. Then the so-called "*semi-gradient*" of Eq. (4) is used to compute the update,

$$\theta_{t+1} = \theta_t - \eta\Phi^\top \boldsymbol{D}\big(\Phi\theta_t - \big(r + \gamma\hat{P}\Phi\theta_t\big)\big)\,, \tag{6}$$

where $\eta$ is the learning rate. From Eq. (6), it is clear that in the underparameterized ($d < |\mathrm{supp}(\hat{\mu})|$) regime, if the system converges, it must converge to parameters $\theta^*_{\boldsymbol{D}}$ such that

$$\Phi^\top \boldsymbol{D} r - \Phi^\top \boldsymbol{D}(\Phi - \gamma \hat{P}\Phi)\theta^*_{\boldsymbol{D}} = 0 \quad \Rightarrow \quad \theta^*_{\boldsymbol{D}} = (\Phi^\top \boldsymbol{D}(\Phi - \gamma\hat{P}\Phi))^{-1}\Phi^\top \boldsymbol{D} r\,, \tag{7}$$

where $\theta^*_{\boldsymbol{D}}$ is the *TD fixed point*. That is, given limited representational power, the TD fixed point minimizes the squared projected Bellman error (MSPBE) by solving the *projected Bellman equation*:

$$\Phi\theta^*_{\boldsymbol{D}} = \Pi^{\boldsymbol{D}}_{\Phi}\left(r + \gamma\hat{P}\Phi\theta^*_{\boldsymbol{D}}\right)\,, \tag{8}$$

such that $\Pi^{\boldsymbol{D}}_{\Phi} = \Phi(\Phi^\top \boldsymbol{D}\Phi)^{-1}\Phi^\top \boldsymbol{D}$ is a weighted projection matrix. It is well-known that TD(0) can diverge if the data sampling distribution $\mu$ is not the stationary distribution of the Markov process. One can still compute the TD fixed point directly using batch data, for example using the LSTD algorithm (Bradtke and Barto, 1996), but this requires computation on the order of $O(d^2)$ compared to $O(d)$ of the iterative update algorithm Eq. (6). The value prediction error of TD is discussed in (Tsitsiklis and Van Roy, 1997; Kolter, 2011; Dann et al., 2014; Bhandari et al., 2018).

**Fitted Value Iteration (FVI)** FVI iteratively updates the weight vector by solving a regression problem where the target is constructed from the current estimate (Ernst et al., 2005; Dann et al., 2014), which is also known as approximate dynamic programming (Sutton and Barto, 2018). In particular, given the current weight $\theta_t$ at iteration $t$, the objective Eq. (9) is minimized to obtain $\theta_{t+1}$,

$$\mathrm{FVI}_t(\theta) = \tfrac{1}{2}\left\|r + \gamma\hat{P}\Phi\theta_t - \Phi\theta\right\|^2_{\boldsymbol{D}}\,. \tag{9}$$

A simple calculation shows the TD fixed point matches the fixed point of FVI whenever $\theta_0$ is in the row-span of $D\Phi$. Although convergence of FVI can be established under strong conditions (Szepesvári and Munos, 2008), the algorithm can be quite unstable in the general batch setting (Chen and Jiang, 2019; Wang et al., 2021b).

## 4 Over-Parameterized Linear Value Function Approximation

In this section, we study the convergence properties of the value estimation algorithms introduced in Section 3.1 in the *overparameterized* regime where $d > |\mathrm{supp}(\hat{\mu})|$. To faciliate analysis, we first introduce additional notation to simplify the derivations. Let $\{x_i\}_{i=1}^k$ denote the states in the support of $\hat{\mu}$, such that $n(x_i) > 0$ for all $i = \{1, \dots, k\}$ and $k = |\mathrm{supp}(\hat{\mu})|$. Define a *mask matrix* $\boldsymbol{H} \in \mathbb{R}^{k\times|\mathcal{S}|}$ and a *truncated empirical data distribution matrix* $\boldsymbol{D}_k \in \mathbb{R}^{k\times k}$ according to

$$\boldsymbol{H} = \begin{bmatrix} \mathbf{1}^\top_{x_1} \\ \vdots \\ \mathbf{1}^\top_{x_k}\end{bmatrix}\,, \qquad \boldsymbol{D}_k = \begin{bmatrix} \hat{\mu}(x_1) & & \\ & \ddots & \\ & & \hat{\mu}(x_k)\end{bmatrix}\,, \tag{10}$$

where $\mathbf{1}_{x_i} \in \{0,1\}^{|\mathcal{S}|}$ is an indicator vector such that $\phi(x_i) = \Phi^\top \mathbf{1}_{x_i}$. We can then translate between the full distribution and its support via the following.

**Proposition 1.** *The empirical data distribution matrix $\boldsymbol{D}$ can be decomposed as $\boldsymbol{D} = \boldsymbol{H}^\top \boldsymbol{D}_k \boldsymbol{H}$.*

Let $\boldsymbol{M} = \boldsymbol{H}\Phi$, $\boldsymbol{N} = \boldsymbol{H}\hat{P}\Phi$ and $\boldsymbol{R} = \boldsymbol{H}r$ denote the state features, the expected next state features under the empirical transitions, and the rewards on the support of the data distribution respectively.

**Overparameterized Residual Minimization** We first study the convergence of RM given a fixed $\boldsymbol{D}$. First note that the update Eq. (5) can be re-written as

$$\theta_{t+1} = (\boldsymbol{I}_d - \eta(\boldsymbol{M} - \gamma\boldsymbol{N})^\top \boldsymbol{D}_k(\boldsymbol{M} - \gamma\boldsymbol{N}))\theta_t + \eta(\boldsymbol{M} - \gamma\boldsymbol{N})^\top \boldsymbol{D}_k \boldsymbol{R}\,. \tag{11}$$

In the overparameterized regime, one can easily verify that there are infinitely many solutions $\theta \in \mathbb{R}^d$ satisfying $(\boldsymbol{M} - \gamma\boldsymbol{N})\theta = \boldsymbol{R}$. The gradient of Eq. (11) is zero at any of these solutions, which implies that RM can have infinitely many fixed points. However, given that RM minimizes the MSBE objective via gradient descent, as we show in the following theorem, the RM update initialized from $\theta_0 = 0$ will converge to a unique fixed point.

**Theorem 1.** *With $\eta \le \frac{1}{(1+\gamma)^2}$ and starting from $\theta_0 = 0$, RM converges to $\theta_{RM} = (\boldsymbol{M} - \gamma\boldsymbol{N})^\dagger \boldsymbol{R}$.*

**Remark 1.** *For simplicity we present the fixed points of RM and TD starting from $\theta_0 = 0$. The fixed points given an arbitrary initial weight vector $\theta_0 \in \mathbb{R}^d$ are shown in Appendices A.1 and A.2.*

This result parallels similar findings in the supervised learning literature, that training overparameterized deep models with gradient descent (or related algorithms) encodes implicit regularization

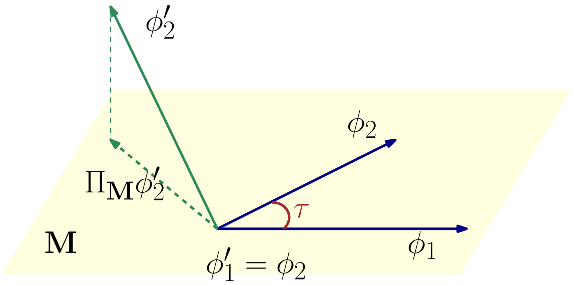


Figure 1: An illustrative example showing the spectrum of $\boldsymbol{W}$ with $k = 2$ and $d = 3$. $\boldsymbol{M} = [\phi_1, \phi_2]^\top$. Without loss of generality, we let $\phi_1 = (\cos\tau, \sin\tau, 0)$ and $\phi_2 = (1, 0, 0)$. $\boldsymbol{N} = [\phi_1', \phi_2']^\top$, where $\phi_1' = \phi_2$, $\phi_2' = (-\cos\tau, \frac{\sqrt{2}}{2}\sin\tau, \frac{\sqrt{2}}{2}\sin\tau)$. Then $\boldsymbol{W} = [[0,1]^\top, [\frac{\sqrt{2}}{2}, -(1 + \frac{\sqrt{2}}{2}\cos\tau)]^\top]$. Clearly, the spectral norm of $\boldsymbol{W}$ increases as the angle $\tau$ between $\phi_1$ and $\phi_2$ decreases.

that drives the model solution to particular outcomes in the overparameterized regime (Soudry et al., 2018; Gunasekar et al., 2018; Neyshabur et al., 2019). Moreover, this implicit regularization is often associated with generalization benefits. However, unlike the case for supervised learning, RM solutions do not often generalize well. Below we uncover a key difference between the RM fixed point and those of TD and FVI that sheds new light on the source of generalization differences.

**Overparameterized TD Learning** We next consider the convergence properties of the TD(0) update in the overparameterized setting. First, rewrite the TD(0) update formula Eq. (6) as

$$\theta_{t+1} = (\boldsymbol{I}_d - \eta \boldsymbol{M}^\top \boldsymbol{D}_k(\boldsymbol{M} - \gamma \boldsymbol{N}))\theta_t + \eta \boldsymbol{M}^\top \boldsymbol{D}_k \boldsymbol{R}\,. \tag{12}$$

Similar to RM, in the overparameterized regime any solutions $\theta \in \mathbb{R}^d$ that satisfy $(\boldsymbol{M} - \gamma\boldsymbol{N})\theta = \boldsymbol{R}$ are the fixed points of Eq. (12), which implies an infinite set of fixed points. This is quite unlike the underparameterized case where there is a unique TD fixed point Eq. (7) given by the solution of projected Bellman equation. However, we now show that in the overparameterized setting, similar to solving RM using gradient descent, TD also encodes an implicit bias toward a particular fixed point.

This of course requires TD to converge, which can be assured by a simple condition. Let $\boldsymbol{W} = \boldsymbol{N}\boldsymbol{M}^\dagger$, which has a geometric interpretation that we will exploit later in Section 5. Observe that

$$\boldsymbol{N} = \boldsymbol{N}\Pi_{\boldsymbol{M}} + \boldsymbol{N}(\boldsymbol{I}_d - \Pi_{\boldsymbol{M}}) = \boldsymbol{N}\boldsymbol{M}^\dagger\boldsymbol{M} + \boldsymbol{N}(\boldsymbol{I}_d - \Pi_{\boldsymbol{M}}) = \boldsymbol{W}\boldsymbol{M} + \boldsymbol{N}(\boldsymbol{I}_d - \Pi_{\boldsymbol{M}})\,, \tag{13}$$

i.e., $\boldsymbol{N}$ can be decomposed into its projection onto the row-span of $\boldsymbol{M}$ plus a perpendicular component. Eq. (13) shows that $\boldsymbol{W}$ projects $\boldsymbol{N}$ onto the row space of $\boldsymbol{M}$; see Fig. 1 for an illustration. We refer to $\boldsymbol{W}$ as the *core matrix* since its norm determines the convergence of TD.

**Theorem 2.** *Choosing $\eta < \frac{1}{(1+\gamma)\|\Phi\|}$ and starting from $\theta_0 = 0$, if $\|\boldsymbol{W}\| < \frac{1}{\gamma}$, TD(0) converges to $\theta_{TD} = \boldsymbol{M}^\dagger(\boldsymbol{I}_k - \gamma\boldsymbol{W})^{-1}\boldsymbol{R}$. If $\|\boldsymbol{W}\| \geq \frac{1}{\gamma}$ there is an initial $\theta_0$ for which TD(0) does not converge.*

A few key observations. First, note that the RM fixed point in Theorem 1 and the TD fixed point in Theorem 2 are not identical. That is, the different value estimation algorithms continue to demonstrate different preferences for fixed points, but in the overparameterized setting these differences are *implicit* in the algorithms and cannot be captured by the MSBE versus MSPBE objectives, since both are zero for any $\theta$ that satisfies $(\boldsymbol{M} - \gamma\boldsymbol{N})\theta = \boldsymbol{R}$. Second, *the fixed point of TD lies in different basis than RM*. That is, $\theta_{\text{TD}}$ lies in the row space of the state features $\boldsymbol{M}$, whereas $\theta_{\text{RM}}$ lies in the row space of the residual features $\boldsymbol{M} - \gamma\boldsymbol{N}$, and these two spaces are not identical in general. We revisit the significance of this difference below, but intuitively, the parameter vector $\theta$ is being trained to predict values rather than temporal differences, and the future test states from which value predictions are made will tend to be closer to the space spanned by $\boldsymbol{M}$ than $\boldsymbol{M} - \gamma\boldsymbol{N}$.

**Overparameterized Fitted Value Iteration** Finally, we consider the convergence of FVI. Recall that at iteration $t$, FVI solves the least squares problem Eq. (9) to compute the next weight vector. Using the notation established above, the normal equations for this problem can be expressed as $\boldsymbol{M}^\top\boldsymbol{D}_k\boldsymbol{M}\theta = \boldsymbol{M}^\top\boldsymbol{D}_k(\boldsymbol{R} + \gamma\boldsymbol{N}\theta_t)$, but this system cannot be directly used to compute the solution since $\boldsymbol{M}^\top\boldsymbol{D}_k\boldsymbol{M}$ is not invertible. Furthermore, just like RM and TD, any $\theta \in \mathbb{R}^d$ that satisfies $(\boldsymbol{M} - \gamma\boldsymbol{N})\theta = \boldsymbol{R}$ is a fixed point of FVI. If one solves the least squares problem Eq. (9) using gradient descent, it is known (Bartlett et al., 2021; Soudry et al., 2018) that the optimization converges to the minimum norm solution

$$\theta_{t+1} = \boldsymbol{M}^\dagger(\boldsymbol{R} + \gamma\boldsymbol{N}\theta_t)\,. \tag{14}$$

Interestingly, by choosing this solution, each iteration of FVI corresponds to applying a linear backup on the current value estimate, where the backup operator is defined by the core matrix.

**Definition 1.** *Define the core matrix linear operator $\mathcal{T}_{\boldsymbol{W}}$ by $\mathcal{T}_{\boldsymbol{W}}\nu = \boldsymbol{R} + \gamma \boldsymbol{W}\nu$ for any $\nu \in \mathbb{R}^{|\mathcal{S}|}$.*

Using this operator we can characterize the convergence condition of FVI, reaching the conclusion that whenever $\mathcal{T}_{\boldsymbol{W}}$ is a non-expansion, FVI converges to the same fixed point as TD.

**Theorem 3.** *Let $\theta_0$ be the initial weight and $\theta_t \in \mathbb{R}^d$ be the output of FVI at iteration $t$. We have*

$$\theta_{t+1} = \boldsymbol{M}^{\dagger}\mathcal{T}_{\boldsymbol{W}}^{t}(\boldsymbol{R} + \gamma \boldsymbol{N}\theta_0)\,. \tag{15}$$

*Furthermore, given that $\|\boldsymbol{W}\| < 1/\gamma$, the algorithm converges to $\theta_{TD} = \boldsymbol{M}^{\dagger}(\boldsymbol{I}_k - \gamma\boldsymbol{W})^{-1}\boldsymbol{R}$. If $\|\boldsymbol{W}\| \geq \frac{1}{\gamma}$ there is an initial $\theta_0$ for which FVI does not converge.*

### 4.1 Unified View of Overparameterized Value Estimators

We now show that the convergence points above can be characterized as solutions to related constrained optimization problems, providing a unified perspective on the respective algorithm biases.

**Theorem 4.** *$\theta_{RM}$ is the solution of the following constrained optimization,*

$$\inf_{\theta\in\mathbb{R}^d} \tfrac{1}{2}\|\theta\|^2 \quad \textit{s.t. } \boldsymbol{M}\theta = \boldsymbol{R} + \gamma\boldsymbol{N}\theta\,, \tag{16}$$

*and $\theta_{TD}$ is the solution of the following constrained optimization,*

$$\inf_{\theta\in\mathbb{R}^d} \tfrac{1}{2}\|\theta\|^2 \quad \textit{s.t. } \boldsymbol{M}\theta = \boldsymbol{R} + \gamma\boldsymbol{N}\theta\,,\ \mathrm{null}(\boldsymbol{M})\theta = 0\,. \tag{17}$$

That is, the convergence points of RM, TD and FVI in the overparameterized case can all be seen as minimizing the Euclidean norm of the weights $\theta$ subject to satisfying the Bellman constraints $\boldsymbol{M}\theta = \boldsymbol{R} + \gamma\boldsymbol{N}\theta$, where TD and FVI implicitly add the additional constraint that $\theta$ must lie in the row span of $\boldsymbol{M}$; moreover, this is the *only* constraint that differentiates TD from RM. From this perspective, the algorithms can all be seen as iterative procedures for solving a particular form of quadratic program, when they converge. Of course, proper constrained optimization techniques would be able to stably compute solutions in scenarios where TD or FVI diverge (Boyd and Vandenberghe, 2004), but a more direct way to ensure convergence is implied by the following corollary.

**Corollary 1.** *$\theta_{TD}$ is also the solution of the following constrained optimization,*

$$\inf_{\theta\in\mathbb{R}^d} \tfrac{1}{2}\|\theta\|^2 \quad \textit{s.t. } \boldsymbol{M}\theta = \boldsymbol{R} + \gamma\boldsymbol{N}\Pi_{\boldsymbol{M}}\theta\,. \tag{18}$$

Note that the right hand side of the constraint simply pre-projects next state value predictions onto the row space of $\boldsymbol{M}$ before determining the Bellman backed up value. This allows a novel objective to be formulated whose minimizer recovers the same fixed point as TD,

$$\mathrm{MSCBE}(\theta) = \tfrac{1}{2}\left\|\boldsymbol{R} + \gamma\boldsymbol{N}\Pi_{\boldsymbol{M}}\theta - \boldsymbol{M}\theta\right\|_{\boldsymbol{D}}^2\,, \tag{19}$$

which stands for mean squared *corrected* Bellman error. Note that MSCBE is not identical to MSPBE because the projection is applied before not after the Bellman backup. Gradient descent minimization of MSCBE yields the same fixed point as $\theta_{\mathrm{TD}}$, which is essentially equivalent to applying RM to corrected target values while ensuring stability. Note also that in the linear case the projection matrix $\Pi_{\boldsymbol{M}}$ only needs to be precomputed once.

### 4.2 Value Prediction Error Bounds

One can also establish generalization bounds on the value estimation error of these methods in the overparameterized regime. We first provide a finite time analysis of the value prediction error of FVI.

**Theorem 5.** *Let $\hat{\Sigma} = \boldsymbol{M}^{\top}\boldsymbol{D}_k\boldsymbol{M}$ be the empirical covariance matrix, and $\theta_t$ be the output of FVI starting from $\theta_0$ as defined in Theorem 3. Then for any $\theta^* \in \arg\min_{\theta\in\mathbb{R}^d}\mathcal{E}(\theta)$,*

$$\begin{aligned}&\mathcal{E}(\theta_t) - \mathcal{E}(\theta^*)\\ &\leq \tfrac{1}{k\lambda_{\min}(\hat{\Sigma})}\left(\left(\varepsilon^2 + \sigma^2\right)\left\|\sum_{i=0}^{t-1}(\gamma\boldsymbol{W})^i\right\|^2 + \left\|(\gamma\boldsymbol{W})^{t-1}\right\|^2\|\Phi\|^2\|\theta_0 - \theta^*\|^2\right) + \tfrac{1}{2}\left\|\theta^*\right\|_{\boldsymbol{I}_d - \Pi_{\boldsymbol{M}}}^2\,,\end{aligned} \tag{20}$$

*where $\varepsilon = \|\boldsymbol{N}(\boldsymbol{I}_d - \Pi_{\boldsymbol{M}})\theta^*\|$ and $\sigma = \|\boldsymbol{H}(\hat{P} - P)v\|$.*

Intuitively, $\varepsilon$ measures the length of next-state features along the direction $\theta^*$, and $\sigma$ is the expected value prediction error under the empirical transition model, which can be bounded using standard concentration inequalities. The proof of this theorem is given in Appendix A.5. Observe that for any step $t \geq 1$, the output of FVI $\theta_t$ is within the row-span of $\boldsymbol{M}$. This allows us to decompose the prediction error into a component within the row-span, controlled by leveraging the core matrix linear operator $\mathcal{T}_{\boldsymbol{W}}$, and an orthogonal component that can be bounded by $\|\theta^*\|^2_{\boldsymbol{I}_d - \Pi_{\boldsymbol{M}}}$.

Under the convergence conditions of Theorems 2 and 3, we also have the following generalization bound for the value prediction error of $\theta_{\text{TD}}$.

**Corollary 2.** *Suppose that $\|\boldsymbol{W}\| \leq 1$, and the value prediction for any $s \in \mathcal{S}$ is bounded by $v(s) \in [0, v_{\max}]$. For any $\theta^* \in \arg\min_{\theta \in \mathbb{R}^d} \mathcal{E}(\theta)$,*

$$\mathbb{E}[\mathcal{E}(\theta_{\text{TD}})] \leq \frac{\gamma \log(|\mathcal{S}|/\delta)}{n_{\min}\mathbb{E}[\lambda_{\min}(\hat{\Sigma})](1-\gamma)^4} + \frac{4\gamma \mathbb{E}[\|\theta^*\|^2_{\boldsymbol{I}_d - \Pi_{\boldsymbol{M}}}]}{\mathbb{E}[\lambda_{\min}(\hat{\Sigma})](1-\gamma)^2} + \delta v_{\max} \,, \tag{21}$$

*where $n_{\min} = \min_{s:n(s)>0} n(s)$ is the minimum counts given the data set.*

This result automatically implies the requirements for ensuring offline generalization, accounting both for distribution shift (Wang et al., 2021b) and policy completeness (Munos and Szepesvári, 2005; Duan et al., 2020) in feature space. In particular, for Eq. (20) and Eq. (21), we characterize the distribution shift using well known concentration bounds in Appendix A.6, which leads to the denominators $k\lambda_{\min}(\hat{\Sigma})$ and $n_{\min}\mathbb{E}[\lambda_{\min}(\hat{\Sigma})]$ respectively. In addition, we explicitly characterize the misalignment between the features of current states and next states using the core matrix, which can be used to bound misalignment between values, replacing the feature completeness assumption.

We note that if the convergence condition cannot be satisfied, that is when $\|\boldsymbol{W}\| \geq 1/\gamma$, the estimation error could be arbitrarily large. The sources of value estimation error are explicit in Corollary 2. The first term measures the error due to sampling (statistical error), while the second term considers out-of-span components of the optimal weight vector $\theta^*$ with respect to $\boldsymbol{M}$ (approximation error). The smallest eigenvalue of the empirical covariance matrix $\mathbb{E}[\lambda_{\min}(\hat{\Sigma})]$, as well as the length of the orthogonal components $\mathbb{E}[\|\theta^*\|^2_{\boldsymbol{I}_d - \Pi_{\boldsymbol{M}}}]$, can both be controlled using classical techniques for concentration properties of random matrix. In Appendix A.7 we present the exact approach for bounding these two terms. Furthermore, by Corollary 1, one can also apply Corollary 2 to an algorithm that directly optimizes MSCBE. Although a solution of Eq. (18) must exist, its value prediction error can be arbitrarily large given that $\|\boldsymbol{W}\| \geq 1/\gamma$. This also connects to a similar result for the TD fixed point that minimizes MSPBE in the underparameterized regime (Kolter, 2011).

## 5 Regularizers for Deep Reinforcement Learning Algorithms

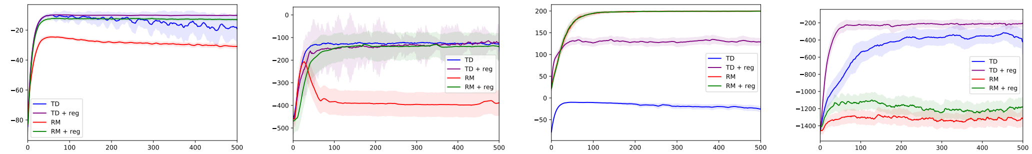
For tractability, the theory in prior sections assumes fixed representations with a linear parameterization on only the final layer parameters of the value function. However, in practice, deep RL algorithms also learn the representations in an end-to-end fashion. Inspired by the linear case, we now identify two novel regularizers that are applicable more generally—one that closes the gap between RM and TD inspired by the unified view of different fixed points, and another that quantifies the effect of feature representation on the generalization bound.

**Two-Part Approximation** Most deep RL algorithms rely on approximating the value function with a deep neural network $Q_\omega$ that predicts the future outcome of given state-action pair (Mnih et al., 2015; Kalashnikov et al., 2018; Lillicrap et al., 2016). In practice, $Q_\omega$ is trained by TD learning that minimizes the objective $\sum_{s,a}(r(s,a) + \gamma\bar{Q}_\omega(s,a) - Q_\omega(s,a))$, where $\bar{Q}_\omega(s,a)$ is known as the *target network* to increase the learning stability. We view $Q_\omega$ as a two part-approximation with $\omega = (\phi, \theta)$, where the output of the penultimate layer is referred as the feature mapping $\phi$, the weight of last fully connected layer is referred as $\theta$, and the Q-function is approximated by $Q_\omega(s,a) = \phi(s,a)^\top\theta$. Our goal is to define regularizers on $\phi$ and $\theta$ that can be effectively applied to practical algorithms.

The first regularizer directly takes inspiration from Theorem 4: by restricting the linear weight $\theta$ within the row space of $\boldsymbol{M}$ (now defined by exited $(s,a)$ pairs in the data), RM finds the same fixed point as TD. We implement this idea by penalizing the norm of the perpendicular component of $\theta$,

$$\mathcal{R}_\theta = \|\theta - \Pi_{\boldsymbol{M}}\theta\| \,, \tag{22}$$

In practice we compute this regularizer for each minibatch of data. The projection step is computed by a least squares algorithm with an additional $l_2$ regularization for numerical stability.



(a) Reacher (b) Acrobot (c) Cartpole (d) Pendulum

Figure 2: We show the results with proposed regularization compared to the baseline algorithms. The algorithms are trained using a fixed offline data set collected by random initialized policies. The $x$-axis shows the training iterations (in thousands) and $y$-axis shows the performance. All plots are averaged over 100 runs. The shaded area shows the standard error.

The second regularizer is designed to address the effect of the feature representation on convergence and value prediction error. In particular, as shown by Theorems 2 and 3, a sufficient condition that guarantees the convergence of TD and FVI is that the spectral norm of $\boldsymbol{W}$ be upper bounded by $1/\gamma$, which by Theorem 5 will also reduce the bound on generalization error. Hence, it is natural to penalize the norm of this matrix using standard automatic differentiation tools. However, such an approach is prone to numerical difficulty, as it involves differentiation through a matrix pseudo inverse. We instead propose an alternative regularizer inspired by the geometric interpretation of the core matrix Eq. (13): recall from Fig. 1 that $\boldsymbol{W}$ can be viewed as the weights that project $\boldsymbol{N}$ onto the row space of $\boldsymbol{M}$. To ensure that an arbitrary feature vector can be well approximated using $\boldsymbol{W}$, it would be ideal if $\boldsymbol{M}$ was orthonomral, which would imply an ideally-behaved basis to represent $\boldsymbol{N}$. This intuition justifies the following regularization:

$$\mathcal{R}_\phi = \left\| \beta \boldsymbol{I}_d - \boldsymbol{M}^\top \boldsymbol{D}_k \boldsymbol{M} \right\| , \tag{23}$$

where $\beta$ is a scale parameter designed to approximate the column norm. That is, the regularizer forces the neural network to learn an orthogonal feature embedding by normalizing the empirical feature covariance matrix. The gradient of $\mathcal{R}_\phi$ can also approximated using mini-batches. We augment the original learning objectives by adding both $\mathcal{R}_\theta$ and $\mathcal{R}_\phi$ weighted by hyper-parameters.

### 5.1 Empirical Justification of Regularizers

The goal of our experiments is to assess the applicability of the proposed regularization schemes based on orthogonality and projection operations to practical deep RL algorithms. To avoid the confounding effects of exploration, we restrict our study to learning from a frozen batch of data with a fixed number of transitions collected prior to learning. We use a *randomly initialized policy* in this initial collection step. We consider both discrete and continuous control benchmarks in this analysis. For the discrete action environments, we use DQN (Mnih et al., 2015) as the baseline algorithm to add our regularizers. For continuous control environments, we use QT-Opt (Kalashnikov et al., 2018) as the baseline algorithm, which is an actor-critic method that applies the cross-entropy method to perform policy optimization. Our modifications add $\mathcal{R}_\phi$ and $\mathcal{R}_\theta$ to the standard MSBE objective on the critic Q-network. Additional details describing the complete experiment setup for each environment are provided in Appendix B. Experimental results contrasting vanilla TD and RM with their regularized variants are summarized in Fig. 2. These findings demonstrate that our regularization schemes can be used to improve the performance of both vanilla TD learning and RM. Note that RM is typically less popular than TD due to its worse empirical performance. On Acrobot and Reacher, our method was able to fully close the gap between RM and TD. On Cartpole, (where vanilla RM dominates vanilla TD), and on Pendulum, our regularizers also deliver significant improvements to the TD learning baseline and modest improvements to the RM baseline.

## 6 Conclusion

We have investigated the fixed points of classical updates for value estimation in the overparameterized setting, where there is sufficient capacity to fit all the Bellman constraints in a given data set. We find that TD and FVI have different fixed points than RM, but in the linear case the difference can be entirely attributed to a constraint missing from RM that the solution lie in the row space of the predecessor state features. We devised two novel regularizers based on these findings, which stabilized the performance of TD without sacrificing generalization, while improving the generalization of RM, in the setting of estimating optimal values with a deep model. Characterizing the implicit bias of other algorithms, such as gradient or emphatic TD variants remains open. Identifying other regularizers that further close the gap between TD and RM is also an interesting direction for future investigation.

## REFERENCES

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

Alekh Agarwal, Sham Kakade, Akshay Krishnamurthy, and Wen Sun. FLAMBE: Structural complexity and representation learning of low rank MDPs. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, 2020.

Rishabh Agarwal, Max Schwarzer, Pablo Samuel Castro, Aaron Courville, and Marc G. Bellemare. Deep reinforcement learning at the edge of the statistical precipice. arXiv preprint arXiv:2108.13264, 2021.

Sanjeev Arora, Simon S Du, Wei Hu, Zhiyuan Li, Ruslan Salakhutdinov, and Ruosong Wang. On exact computation with an infinitely wide neural net. *arXiv preprint arXiv:1904.11955*, 2019.

Leemon Baird. Residual algorithms: Reinforcement learning with function approximation. In *International Conference on Machine Learning (ICML)*, pages 30–37, 1995.

Peter L Bartlett, Andrea Montanari, and Alexander Rakhlin. Deep learning: a statistical viewpoint. *arXiv preprint arXiv:2103.09177*, 2021.

Mikhail Belkin, Daniel Hsu, Siyuan Ma, and Soumik Mandal. Reconciling modern machine-learning practice and the classical bias-variance trade-off. *Proceedings of the National Academy of Sciences*, 116(32):15849–15854, 2019.

Marc G. Bellemare, Salvatore Candido, Pablo Samuel Castro, Jun Gong, Marlos C. Machado, Subhodeep Moitra, Sameera S. Ponda, and Ziyu Wang. Autonomous navigation of stratospheric balloons using reinforcement learning. *Nature*, 588:77–82, 2020.

Jalaj Bhandari, Daniel Russo, and Raghav Singal. A finite time analysis of temporal difference learning with linear function approximation. In *Conference on Learning Theory (COLT)*, pages 1691–1692, 2018.

Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge U Press, 2004.

Steven J Bradtke and Andrew G Barto. Linear least-squares algorithms for temporal difference learning. *Machine Learning*, 22(1):33–57, 1996.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, 2020.

Diogo Carvalho, Francisco S Melo, and Pedro Santos. A new convergent variant of Q-learning with linear function approximation. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, 2020.

Jinglin Chen and Nan Jiang. Information-theoretic considerations in batch reinforcement learning. In *International Conference on Machine Learning (ICML)*, pages 1042–1051, 2019.

Bo Dai, Niao He, Yunpeng Pan, Byron Boots, and Le Song. Learning from conditional distributions via dual embeddings. In *Artificial Intelligence and Statistics*, pages 1458–1467. PMLR, 2017.

Gal Dalal, Balázs Szörényi, Gugan Thoppe, and Shie Mannor. Finite sample analysis for TD$(0)$ with function approximation. In *AAAI Conference on Artificial Intelligence (AAAI)*, pages 6144–6160, 2018.

Christoph Dann, Gerhard Neumann, and Jan Peters. Policy evaluation with temporal differences: A survey and comparison. *Journal of Machine Learning Research*, 15:809–883, 2014.

Simon S Du, Jianshu Chen, Lihong Li, Lin Xiao, and Dengyong Zhou. Stochastic variance reduction methods for policy evaluation. In *International Conference on Machine Learning*, pages 1049–1058. PMLR, 2017.

Yaqi Duan, Zeyu Jia, , and Mengdi Wang. Minimax-optimal off-policy evaluation with linear function approximation. In *International Conference on Machine Learning (ICML)*, pages 2701–2709, 2020.

Damien Ernst, P. Guerts, and L. Whenkel. Tree-based batch mode reinforcement learning. *Journal of Machine Learning Research*, 6:503–556, 2005.

Mattheiu Geist, Bilal Piot, and Olivier Pietquin. Is the Bellman residual a bad proxy? In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 31, pages 3208–3217, 2017.

Dibya Ghosh and Marc G. Bellemare. Representations for stable off-policy reinforcement learning. In *International Conference on Machine Learning (ICML)*, pages 3556–3565, 2020.

Geoffrey J. Gordon. Stable function approximation in dynamic programming. In *International Conference on Machine Learning (ICML)*, pages 261–268, 1995.

Suriya Gunasekar, Jason Lee, Daniel Soudry, and Nathan Srebro. Characterizing implicit bias in terms of optimization geometry. In *International Conference on Machine Learning (ICML)*, 2018.

Botao Hao, Yaqi Duan, Tor Lattimore, Csaba Szepesvári, and Mengdi Wang. Sparse feature selection makes batch reinforcement learning more sample efficient. In *International Conference on Machine Learning (ICML)*, 2021.

Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 31, pages 8571–8580, 2018.

Chi Jin, Zhuoran Yang, Zhaoran Wang, and Michael I. Jordan. Provably efficient reinforcement learning with linear function approximation. *Journal of Machine Learning Research*, 125:1–7, 2020.

Dmitry Kalashnikov, Alex Irpan, Peter Pastor, Julian Ibarz, Alexander Herzog, Eric Jang, Deirdre Quillen, Ethan Holly, Mrinal Kalakrishnan, Vincent Vanhoucke, et al. QT-Opt: Scalable deep reinforcement learning for vision-based robotic manipulation. *arXiv preprint arXiv:1806.10293*, 2018.

J. Zico Kolter. The fixed points of off-policy TD. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 24, pages 2169–2177, 2011.

George Konidaris, Scott Niekum, and Philip S. Thomas. TD$_\gamma$: Re-evaluating complex backups in temporal difference learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 24, pages 2402–2410, 2011.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. ImageNet classification with deep neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 25, 2012.

Aviral Kumar, Rishabh Agarwal, Dibya Ghosh, and Sergey Levine. Implicit under-parameterization inhibits data-efficient deep reinforcement learning. In *International Conference on Learning Representations (ICLR)*, 2021.

Ilja Kuzborskij, Csaba Szepesvári, Omar Rivasplata, Amal Rannen-Triki, and Razvan Pascanu. On the role of optimization in double descent: A least squares study. arXiv preprint arXiv:2107.12685, 2021.

Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. In *ICLR*, 2016.

Bo Liu, Ji Liu, Mohammad Ghavamzadeh, Sridhar Mahadevan, and Marek Petrik. Finite-sample analysis of proximal gradient TD. In *Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 504–513, 2015.

Bo Liu, Ji Liu, Mohammad Ghavamzadeh, Sridhar Mahadevan, and Marek Petrik. Proximal gradient temporal difference learning algorithms. In *International Joint Conference on Artificial Intelligence (IJCAI)*, pages 4195–4199, 2016.

Daniel J. Lizotte. Convergent fitted value iteration with linear function approximation. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 24, pages 2537–2545, 2011.

Tyler Lu, Dale Schuurmans, and Craig Boutilier. Non-delusional Q-learning and value iteration. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 32, pages 9971–9981, 2018.

Hamid R. Maei, Csaba Szepesvári, Shalabh Bhatnagar, Doina Precup, and David Silver. Convergent temporal-difference learning with arbitrary smooth function approximation. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 22, pages 1204–1212, 2009.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin Riedmiller, Andreas K. Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015.

Rémi Munos and Csaba Szepesvári. Finite time bounds for sampling based fitted value iteration. In *International Conference on Machine Learning (ICML)*, pages 880–887, 2005.

Behnam Neyshabur, Zhiyuan Li, Srinadh Bhojanapalli, Yann LeCun, and Nathan Srebro. The role of over-parameterization in generalization of neural networks. In *International Conference on Learning Representations (ICLR)*, 2019.

Ronald Parr, Lihong Li, Gavin Taylor, Christopher Painter-Wakefield, and Michael L. Littman. An analysis of linear models, linear value-function approximation, and feature selction for reinforcement learning. In *International Conference on Machine Learning (ICML)*, pages 752–759, 2008.

Gandharv Patil, L. A. Prashanth, and Doina Precup. Finite time analysis of temporal difference learning with linear function approximation: the tail averaged case. 2021.

L. A. Prashanth, Nathaniel Korda, and Rémi Munos. Concentration bounds for temporal difference learning with linear function approximation: The case of batch data and uniform sampling. *Machine Learning*, 110(3):559–618, 2021.

Bruno Scherrer. Should one compute the temporal difference fix point or minimize the Bellman residual? In *International Conference on Machine Learning (ICML)*, 2010.

John Shawe-Taylor, Christopher KI Williams, Nello Cristianini, and Jaz Kandola. On the eigenspectrum of the Gram matrix and the generalization error of kernel-PCA. *IEEE Transactions on Information Theory*, 51(7):2510–2522, 2005.

Zhao Song, Ronald Parr, Xuejun Liao, and Lawrence Carin. Linear feature encoding for reinforcement learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2016.

Daniel Soudry, Elad Hoffer, Mor Shpigel Nacson, Suriya Gunasekar, and Nathan Srebro. The implicit bias of gradient descent on separable data. *Journal of Machine Learning Research*, 19:1–57, 2018.

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.

Richard S Sutton, David McAllester, Satinder Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 12, 1999.

Richard S Sutton, Csaba Szepesvári, and Hamid Reza Maei. A convergent $O(n)$ algorithm for off-policy temporal-difference learning with linear function approximation. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 21, pages 1609–1616, 2008.

Richard S. Sutton, A. Rupam Mahmood, and Martha White. An emphatic approach to the problem of off-policy temporal-difference learning. *Journal of Machine Learning Research*, 17:1–29, 2016.

Csaba Szepesvári and Rémi Munos. Finite-time bounds for fitted value iteration. *Journal of Machine Learning Research*, 9(5), 2008.

Csaba Szepesvári and W. D. Smart. Interpolation-based Q-learning. In *International Conference on Machine Learning (ICML)*, pages 100–107, 2004.

Gavin Taylor and Ronald Parr. Kernelized value function approximation for reinforcement learning. In *International Conference on Machine Learning (ICML)*, pages 4214–4226, 2009.

John N Tsitsiklis and Benjamin Van Roy. Feature-based methods for large scale dynamic programming. *Machine Learning*, 22(1):59–94, 1996.

John N Tsitsiklis and Benjamin Van Roy. An analysis of temporal-difference learning with function approximation. *IEEE Transactions on Automatic Control*, 42(5):674–690, 1997.

Hado van Hasselt. Double Q-learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 23, 2010.

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.

Ruosong Wang, Dean Foster, and Sham M Kakade. What are the statistical limits of offline RL with linear function approximation? In *International Conference on Learning Representations (ICLR)*, 2021a.

Ruosong Wang, Yifan Wu, Ruslan Salakhutdinov, and Sham M. Kakade. Instabilities of offline RL with pre-trained neural representation. In *International Conference on Machine Learning (ICML)*, pages 10948–10960, 2021b.

Wentao Weng, Harsh Gupta, Niao He, and Lei Ying. The mean-squared error of double Q-learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Chenjun Xiao, Ilbin Lee, Bo Dai, Dale Schuurmans, and Csaba Szepesvari. On the sample complexity of batch reinforcement learning with policy-induced data. *arXiv preprint arXiv:2106.09973*, 2021.

Lin F. Yang and Mengdi Wang. Sample-optimal parametric Q-learning using linearly additive features. In *International Conference on Machine Learning (ICML)*, 2019.

Zhuoran Yang, Chi Jin, Zhaoran Wang, Mengdi Wang, and Michael I. Jordan. On function approximation in reinforcement learning: Optimism in the face of large state spaces. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Huizhen Yu. Convergence of least squares temporal difference methods under general conditions. In *International Conference on Machine Learning (ICML)*, pages 1207–1214, 2010.

Huizhen Yu. On convergence of emphatic temporal-difference learning. In *Conference on Learning Theory (COLT)*, pages 1–28, 2015.

Andrea Zanette. Exponential lower bounds for batch reinforcement learning. In *International Conference on Machine Learning (ICML)*, pages 12287–12297, 2021.

Shangtong Zhang, Hengshuai Yao, and Shimon Whiteson. Breaking the deadly triad with a target network. In *International Conference on Machine Learning (ICML)*, pages 12621–12631, 2021.

# Appendix

## A Proofs

### A.1 Proof of Theorem 1

**Theorem 6** (Restatement of Theorem 1)**.** *Let $\theta_0 \in \mathbb{R}^d$ be the initial weight vector. With $\eta \leq \frac{1}{(1+\gamma)^2}$, RM converges to $\theta_{\mathrm{RM}} = (\boldsymbol{M} - \gamma\boldsymbol{N})^\dagger \boldsymbol{R} + (\boldsymbol{I}_d - \Pi_{\boldsymbol{M}-\gamma\boldsymbol{N}})\theta_0$.*

*Proof.* Let $\boldsymbol{A} = \boldsymbol{M} - \gamma\boldsymbol{N}$ for simplicity. First recall the residual minimization update,

$$\theta_{t+1} = \left(\boldsymbol{I}_d - \eta\boldsymbol{A}^\top\boldsymbol{D}_k\boldsymbol{A}\right)\theta_t + \eta\boldsymbol{A}^\top\boldsymbol{D}_k\boldsymbol{R}\,. \tag{24}$$

Let $\theta^* = \boldsymbol{A}^\dagger\boldsymbol{R}$. It can be verified $\theta^*$ is one of the feasible solution as $\boldsymbol{A}\theta^* = \boldsymbol{R}$. Then a simple recursive argument shows that for any $\theta_0 \in \mathbb{R}^d$,

$$\begin{aligned}
\theta_{t+1} - \theta^* &= \left(\boldsymbol{I}_d - \eta\boldsymbol{A}^\top\boldsymbol{D}_k\boldsymbol{A}\right)\theta_t + \eta\boldsymbol{A}^\top\boldsymbol{D}_k\boldsymbol{R} - \theta^* && (25)\\
&= \left(\boldsymbol{I}_d - \eta\boldsymbol{A}^\top\boldsymbol{D}_k\boldsymbol{A}\right)\theta_t + \eta\boldsymbol{A}^\top\boldsymbol{D}_k\boldsymbol{A}\theta^* - \theta^* && (26)\\
&= (\boldsymbol{I}_d - \eta\boldsymbol{A}^\top\boldsymbol{D}_k\boldsymbol{A})(\theta_t - \theta^*) && (27)\\
&= \dots && (28)\\
&= (\boldsymbol{I}_d - \eta\boldsymbol{A}^\top\boldsymbol{D}_k\boldsymbol{A})^{t+1}(\theta_0 - \theta^*) && (29)
\end{aligned}$$

Thus,

$$\theta_{t+1} = (\boldsymbol{I}_d - \eta\boldsymbol{A}^\top\boldsymbol{D}_k\boldsymbol{A})^{t+1}\theta_0 + (\boldsymbol{I}_d - (\boldsymbol{I}_d - \eta\boldsymbol{A}^\top\boldsymbol{D}_k\boldsymbol{A})^{t+1})\theta^*\,. \tag{30}$$

We let $\boldsymbol{V}\boldsymbol{\Lambda}\boldsymbol{V}^\top$ be its eigendecomposition of $\boldsymbol{A}^\top\boldsymbol{D}_k\boldsymbol{A}$, which is the empirical covariance matrix of residual features. Let $\boldsymbol{V}_-$ be the null space of $\boldsymbol{V}$. Then

$$\begin{aligned}
&\boldsymbol{I}_d - (\boldsymbol{I}_d - \eta\boldsymbol{A}^\top\boldsymbol{D}_k\boldsymbol{A})^{t+1} && (31)\\
=&\boldsymbol{I}_d - (\boldsymbol{V}\boldsymbol{V}^\top - \eta\boldsymbol{V}\boldsymbol{\Lambda}\boldsymbol{V}^\top + \boldsymbol{V}_-\boldsymbol{V}_-^\top)^{t+1} && (32)\\
=&\boldsymbol{I}_d - (\boldsymbol{V}(\boldsymbol{I}_k - \eta\boldsymbol{\Lambda})\boldsymbol{V}^\top + \boldsymbol{V}_-\boldsymbol{V}_-^\top)^{t+1} && (33)\\
=&\boldsymbol{I}_d - (\boldsymbol{V}_-\boldsymbol{V}_-^\top)^{t+1} - \boldsymbol{V}(\boldsymbol{I}_k - \eta\boldsymbol{\Lambda})^{t+1}\boldsymbol{V}^\top && (34)\\
=&\boldsymbol{V}\boldsymbol{V}^\top - \boldsymbol{V}(\boldsymbol{I}_k - \eta\boldsymbol{\Lambda})^{t+1}\boldsymbol{V}^\top && (35)\\
=&\boldsymbol{V}\left(\boldsymbol{I}_k - (\boldsymbol{I}_k - \eta\boldsymbol{\Lambda})^{t+1}\right)\boldsymbol{V}^\top && (36)
\end{aligned}$$

Let $\lambda_{\max}$ be the largest eigenvalue of $\boldsymbol{A}^\top\boldsymbol{D}_k\boldsymbol{A}$. We now show that $\lambda_{\max} \leq 1 + \gamma$.

$$\lambda_{\max}\left(\boldsymbol{A}^\top\boldsymbol{D}_k\boldsymbol{A}\right) \leq \sum_{i=1}^{k}\hat{\mu}(s_i)\lambda_{\max}\Big((\phi(s_i) - \gamma\bar{\phi}(s_i'))(\phi(s_i) - \gamma\bar{\phi}(s_i'))^\top\Big) \leq (1+\gamma)^2\,, \tag{37}$$

where we use the fact that $\lambda_{\max}$ is a convex function and we assume $\|\phi(s)\| \leq 1$ for all $s \in \mathcal{S}$. Thus, given that $\eta \leq \frac{1}{1+\gamma}$, $\eta \leq \frac{1}{\lambda_{\max}}$. Then $\boldsymbol{I}_d - (\boldsymbol{I}_d - \eta\boldsymbol{A}^\top\boldsymbol{D}_k\boldsymbol{A})^{t+1} = \boldsymbol{V}\boldsymbol{V}^\top$ as $t \to \infty$. Thus

$$\lim_{t\to\infty}\theta_t = \lim_{t\to\infty}(\boldsymbol{I}_d - \eta\boldsymbol{A}^\top\boldsymbol{D}_k\boldsymbol{A})^{t+1}\theta_0 + \boldsymbol{V}\boldsymbol{V}^\top\theta^* = \lim_{t\to\infty}(\boldsymbol{I}_d - \eta\boldsymbol{A}^\top\boldsymbol{D}_k\boldsymbol{A})^{t+1}\theta_0 + \theta^*\,, \tag{38}$$

where the last equality follows by that $\theta^*$ is in the row space of $\boldsymbol{A}$ by definition. When $\theta_0 = 0$, we have the algorithm converges to $\theta^*$.

We next show the result for general $\theta_0$. Let $\theta_0 = \theta_0^1 + \theta_0^2$, where $\theta_0^1 = \Pi_{\boldsymbol{M}-\gamma\boldsymbol{N}}\theta_0$ is the component of $\theta_0$ that is in the row space of $\boldsymbol{A}$, $\theta_0^2 = (\boldsymbol{I}_d - \Pi_{\boldsymbol{M}-\gamma\boldsymbol{N}})\theta_0$ is the perpendicular residual. Then,

$$\begin{aligned}
&\lim_{t\to\infty}(\boldsymbol{I}_d - \eta\boldsymbol{A}^\top\boldsymbol{D}_k\boldsymbol{A})^{t+1}\theta_0 && (39)\\
=&\lim_{t\to\infty}(\boldsymbol{V}_-\boldsymbol{V}_-^\top + \boldsymbol{V}(\boldsymbol{I}_k - \eta\boldsymbol{\Lambda})^{t+1}\boldsymbol{V}^\top)(\theta_0^1 + \theta_0^2) && (40)\\
=&\theta_0^2 + \lim_{t\to\infty}\boldsymbol{V}(\boldsymbol{I}_k - \eta\boldsymbol{\Lambda})^{t+1}\boldsymbol{V}^\top\theta_0^1 = \theta_0^2\,, && (41)
\end{aligned}$$

where the last step follows by the choice of $\eta$. This finishes the proof.

□

### A.2 Proof of Theorem 2

**Theorem 7** (Restatement of Theorem 2)**.** *Let $\theta_0 \in \mathbb{R}^d$ be the initial weight vector. Assuming that $\boldsymbol{M}^\top \boldsymbol{D}_k(\boldsymbol{M} - \gamma \boldsymbol{N})$ is diagonalizable, with $\eta < \frac{1}{(1+\gamma)\|\Phi\|}$, if $\|\boldsymbol{W}\| < \frac{1}{\gamma}$, TD(0) converges to $\theta_{TD} = \boldsymbol{M}^\dagger(\boldsymbol{I}_k - \gamma\boldsymbol{W})^{-1}\boldsymbol{R} + \beta$, where $\beta = \boldsymbol{Q}_0\boldsymbol{Q}_0^{-1}\theta_0$, $\boldsymbol{Q}_0$ are eigenvectors of $\boldsymbol{M}^\top \boldsymbol{D}_k(\boldsymbol{M} - \gamma\boldsymbol{N})$ with zero eigenvalues. If $\|\boldsymbol{W}\| \geq \frac{1}{\gamma}$ there is an initial $\theta_0$ for which TD(0) does not converge.*

*Proof.* We first rewrite the TD update formulate as

$$\theta_{t+1} = (\boldsymbol{I}_d - \eta\boldsymbol{M}^\top\boldsymbol{D}_k(\boldsymbol{M} - \gamma\boldsymbol{N}))\theta_t + \eta\boldsymbol{M}^\top\boldsymbol{D}_k\boldsymbol{R} \tag{42}$$

A simple recursive argument shows that for any $\theta_0 \in \mathbb{R}^d$,

$$\theta_{t+1} = (\boldsymbol{I}_d - \eta\boldsymbol{M}^\top\boldsymbol{D}_k(\boldsymbol{M} - \gamma\boldsymbol{N}))^{t-1}\theta_0 + \eta\sum_{i=0}^{t}(\boldsymbol{I}_d - \eta\boldsymbol{M}^\top\boldsymbol{D}_k(\boldsymbol{M} - \gamma\boldsymbol{N}))^i\boldsymbol{M}^\top\boldsymbol{D}_k\boldsymbol{R}\,. \tag{43}$$

Then note that

$$\begin{aligned}
&(\boldsymbol{I}_d - \eta\boldsymbol{M}^\top\boldsymbol{D}_k(\boldsymbol{M} - \gamma\boldsymbol{N}))^i\boldsymbol{M}^\top\boldsymbol{D}_k && (44)\\
=&(\boldsymbol{I}_d - \eta\boldsymbol{M}^\top\boldsymbol{D}_k(\boldsymbol{M} - \gamma\boldsymbol{N}))^{i-1}(\boldsymbol{I}_d - \eta\boldsymbol{M}^\top\boldsymbol{D}_k(\boldsymbol{M} - \gamma\boldsymbol{N}))\boldsymbol{M}^\top\boldsymbol{D}_k && (45)\\
=&(\boldsymbol{I}_d - \eta\boldsymbol{M}^\top\boldsymbol{D}_k(\boldsymbol{M} - \gamma\boldsymbol{N}))^{i-1}(\boldsymbol{M}^\top\boldsymbol{D}_k - \eta\boldsymbol{M}^\top\boldsymbol{D}_k(\boldsymbol{M} - \gamma\boldsymbol{N})\boldsymbol{M}^\top\boldsymbol{D}_k) && (46)\\
=&(\boldsymbol{I}_d - \eta\boldsymbol{M}^\top\boldsymbol{D}_k(\boldsymbol{M} - \gamma\boldsymbol{N}))^{i-1}\boldsymbol{M}^\top\boldsymbol{D}_k(\boldsymbol{I}_k - \eta(\boldsymbol{M} - \gamma\boldsymbol{N})\boldsymbol{M}^\top\boldsymbol{D}_k) && (47)\\
=&\ldots && (48)\\
=&\boldsymbol{M}^\top\boldsymbol{D}_k(\boldsymbol{I}_k - \eta(\boldsymbol{M} - \gamma\boldsymbol{N})\boldsymbol{M}^\top\boldsymbol{D}_k)^i && (49)
\end{aligned}$$

By writing $\boldsymbol{N}$ as the projection to the row-span of $\boldsymbol{M}$ and the perpendicular component, we have

$$\begin{aligned}
&(\boldsymbol{M} - \gamma\boldsymbol{N})\boldsymbol{M}^\top && (50)\\
=&(\boldsymbol{M} - \gamma\boldsymbol{N}\boldsymbol{M}^\dagger\boldsymbol{M} - \gamma\boldsymbol{N}(\boldsymbol{I}_d - \boldsymbol{M}^\dagger\boldsymbol{M}))\boldsymbol{M}^\top && (51)\\
=&(\boldsymbol{I}_k - \gamma\boldsymbol{W})\boldsymbol{M}\boldsymbol{M}^\top\,, && (52)
\end{aligned}$$

where the last step follows by $(\boldsymbol{I}_d - \boldsymbol{M}^\dagger\boldsymbol{M})\boldsymbol{M}^\top = 0$. Thus we can rewrite $\theta_{t+1}$ as

$$\theta_{t+1} = (\boldsymbol{I}_d - \eta\boldsymbol{M}^\top\boldsymbol{D}_k(\boldsymbol{M} - \gamma\boldsymbol{N}))^{t-1}\theta_0 + \eta\boldsymbol{M}^\top\boldsymbol{D}_k\sum_{i=0}^{t}(\boldsymbol{I}_k - \eta(\boldsymbol{M} - \gamma\boldsymbol{N})\boldsymbol{M}^\top\boldsymbol{D}_k)^i\boldsymbol{R} \tag{53}$$

$$= (\boldsymbol{I}_d - \eta\boldsymbol{M}^\top\boldsymbol{D}_k(\boldsymbol{M} - \gamma\boldsymbol{N}))^{t-1}\theta_0 + \eta\boldsymbol{M}^\top\boldsymbol{D}_k\sum_{i=0}^{t}(\boldsymbol{I}_k - \eta(\boldsymbol{I}_k - \gamma\boldsymbol{W})\boldsymbol{M}\boldsymbol{M}^\top\boldsymbol{D}_k)^i\boldsymbol{R}\,, \tag{54}$$

Given $\|\boldsymbol{W}\| < 1/\gamma$, we have that all eigenvalues of $\boldsymbol{I}_k - \gamma\boldsymbol{W}$ are positive. Let $\eta < \frac{1}{(1+\gamma)\|\Phi\|}$, then

$$\|\eta(\boldsymbol{I}_k - \gamma\boldsymbol{W})\boldsymbol{M}\boldsymbol{M}^\top\boldsymbol{D}_k\| < \eta\|(\boldsymbol{I}_k - \gamma\boldsymbol{W})\|\|\boldsymbol{M}\boldsymbol{M}^\top\boldsymbol{D}_k\| < 1\,, \tag{55}$$

otherwise the matrix power series diverges. Thus

$$\eta\boldsymbol{M}^\top\boldsymbol{D}_k\sum_{i=0}^{t}(\boldsymbol{I}_k - \eta(\boldsymbol{I}_k - \gamma\boldsymbol{W})\boldsymbol{M}\boldsymbol{M}^\top\boldsymbol{D}_k)^i\boldsymbol{R} = \boldsymbol{M}^\dagger(\boldsymbol{I}_k - \gamma\boldsymbol{W})^{-1}\boldsymbol{R}\,. \tag{56}$$

Therefore, given that $\theta_0 = 0$, we have the algorithm converge to $\boldsymbol{M}^\dagger(\boldsymbol{I}_k - \gamma\boldsymbol{W})^{-1}\boldsymbol{R}$.

We now show the convergence point for an arbitrary $\theta_0$. Let $\boldsymbol{Q}\Lambda\boldsymbol{Q}^{-1}$ be the eigen decomposition of $\boldsymbol{M}^\top\boldsymbol{D}_k(\boldsymbol{M} - \gamma\boldsymbol{N})$. By the low rank structure of this matrix, it has at most $h \leq k$ non-zero eigenvalues. Let $\boldsymbol{Q}_0$ be the eigenvectors with eigenvalue *zero*. Then

$$\lim_{t\to\infty}(\boldsymbol{I}_d - \eta\boldsymbol{M}^\top\boldsymbol{D}_k(\boldsymbol{M}-\gamma\boldsymbol{N}))^t\theta_0 \tag{57}$$

$$=\lim_{t\to\infty}\boldsymbol{Q}(\boldsymbol{I}_d-\eta\Lambda)^t\boldsymbol{Q}^{-1}\theta_0 \tag{58}$$

$$=\boldsymbol{Q}_0\boldsymbol{Q}_0^{-1}\theta_0\,, \tag{59}$$

where the last step follows by the choice of $\eta$.

□

### A.3 Proof of Theorem 3

*Proof.* We first prove the update formula. Recall the FVI update,

$$\theta = \boldsymbol{M}^\dagger(\boldsymbol{R}+\gamma\boldsymbol{N}\theta_{t-1})\,.$$

For $t=1$ the result holds by definition. For $t\geq 2$, a simple recursive argument reveals that for every $\theta_0\in\mathbb{R}^d$,

$$\theta_t = \boldsymbol{M}^\dagger(\boldsymbol{R}+\gamma\boldsymbol{N}\theta_{t-1}) \tag{60}$$

$$= \boldsymbol{M}^\dagger(\boldsymbol{R}+\gamma\boldsymbol{N}(\boldsymbol{M}^\dagger(\boldsymbol{R}+\gamma\boldsymbol{N}\theta_{t-2}))) \tag{61}$$

$$= \boldsymbol{M}^\dagger(\boldsymbol{R}+\gamma\boldsymbol{N}\boldsymbol{M}^\dagger\boldsymbol{R}+\gamma^2\boldsymbol{N}\boldsymbol{M}^\dagger\boldsymbol{N}\theta_{t-2}) \tag{62}$$

$$= \boldsymbol{M}^\dagger(\boldsymbol{R}+\gamma\boldsymbol{N}\boldsymbol{M}^\dagger\boldsymbol{R}+\gamma^2\boldsymbol{N}\boldsymbol{M}^\dagger\boldsymbol{N}\boldsymbol{M}^\dagger(\boldsymbol{R}+\gamma\boldsymbol{N}\theta_{t-3})) \tag{63}$$

$$= \boldsymbol{M}^\dagger(\boldsymbol{R}+\gamma\boldsymbol{N}\boldsymbol{M}^\dagger\boldsymbol{R}+\gamma^2(\boldsymbol{N}\boldsymbol{M}^\dagger)^2\boldsymbol{R}+\gamma^2(\boldsymbol{N}\boldsymbol{M}^\dagger)^2\boldsymbol{N}\theta_{t-3}) \tag{64}$$

$$= \ldots\ldots \tag{65}$$

$$= \boldsymbol{M}^\dagger\left(\sum_{i=0}^{t-2}(\gamma\boldsymbol{N}\boldsymbol{M})^i\boldsymbol{R}+(\gamma\boldsymbol{N}\boldsymbol{M}^\dagger)^{t-1}(\boldsymbol{R}+\gamma\boldsymbol{N}\theta_0)\right) \tag{66}$$

Clearly the convergence of this algorithm depends on the spectral norm of $\boldsymbol{N}\boldsymbol{M}^\dagger$. In particular, given that $\|\boldsymbol{N}\boldsymbol{M}^\dagger\|<1/\gamma$, we have the algorithm converges to

$$\boldsymbol{M}^\dagger(\boldsymbol{I}_k-\gamma\boldsymbol{W})^{-1}\boldsymbol{R} \tag{67}$$

as $t\to\infty$. This finishes the proof. □

### A.4 Proof of Theorem 4 and Corollary 1

*Proof.* We first prove the result for residual minimization fixed point $\theta_{\mathrm{RM}}$. Let $\boldsymbol{A}=\boldsymbol{M}-\gamma\boldsymbol{N}$ for simplicity. We write the Lagrange of the optimization problem,

$$\mathcal{L}(\theta,\alpha) = \inf_{\theta\in\mathbb{R}^d}\sup_{\alpha\in\mathbb{R}^k}\frac{1}{2}\|\theta\|^2+\alpha^\top(\boldsymbol{R}-\boldsymbol{A}\theta) \tag{68}$$

$$= \sup_{\alpha\in\mathbb{R}^k}\frac{1}{2}\|\boldsymbol{A}^\top\alpha\|^2+\alpha^\top\boldsymbol{R}-\alpha^\top\boldsymbol{A}\boldsymbol{A}^\top\alpha \tag{69}$$

$$= \sup_{\alpha\in\mathbb{R}^k}\alpha^\top\boldsymbol{R}-\frac{1}{2}\alpha^\top\boldsymbol{A}\boldsymbol{A}^\top\alpha\,. \tag{70}$$

Solving for $\alpha^*$ and add it to $\theta^*=\boldsymbol{A}^\top\alpha^*$ gives that $\theta^*=\boldsymbol{A}^\dagger\boldsymbol{R}$.

We next prove Corollary 1, which characterizes the TD and FVI fixed point $\theta_{\mathrm{TD}}$. Let $\boldsymbol{W} = \boldsymbol{N}\boldsymbol{M}^\dagger$. We write the Lagrange of the optimization problem,

$$\begin{aligned}\mathcal{L}(\theta, \alpha) &= \inf_{\theta \in \mathbb{R}^d} \sup_{\alpha \in \mathbb{R}^k} \frac{1}{2}\|\theta\|^2 + \alpha^\top(\boldsymbol{R} - (\boldsymbol{I}_k - \gamma\boldsymbol{W})\boldsymbol{M}\theta) &(71)\\ &= \sup_{\alpha \in \mathbb{R}^k} \frac{1}{2}\|\boldsymbol{M}^\top(\boldsymbol{I}_k - \gamma\boldsymbol{W})^\top\alpha\|^2 + \alpha^\top\boldsymbol{R} - \alpha^\top(\boldsymbol{I}_k - \gamma\boldsymbol{W})\boldsymbol{M}\boldsymbol{M}^\top(\boldsymbol{I}_k - \gamma\boldsymbol{W})^\top\alpha &(72)\\ &= \sup_{\alpha \in \mathbb{R}^k} \alpha^\top\boldsymbol{R} - \frac{1}{2}\alpha^\top(\boldsymbol{I}_k - \gamma\boldsymbol{W})\boldsymbol{M}\boldsymbol{M}^\top(\boldsymbol{I}_k - \gamma\boldsymbol{W})^\top\alpha\,. &(73)\end{aligned}$$

Solving for $\alpha^*$ and add it to $\theta^* = \boldsymbol{M}^\top(\boldsymbol{I}_k - \gamma\boldsymbol{W})^\top\alpha^*$ gives that $\theta^* = \boldsymbol{M}^\dagger(\boldsymbol{I}_k - \gamma\boldsymbol{W})^{-1}\boldsymbol{R}$. The second part of Theorem 4 is immediately followed by this.

□

### A.5 Proof of Theorem 5

**Lemma 1.** *Let $\theta_t$ be the output of FVI at iteration $t$ with $\theta_0$ as the initial parameter. We have that $\boldsymbol{M}^\dagger\boldsymbol{M}\theta_t = \theta_t$ for any $t \geq 1$.*

*Proof.* The claim is implied by the fact that $\theta_t$ is in the row space of $\boldsymbol{M}$. In particular, by Theorem 3, $\theta_t = \boldsymbol{M}^\dagger\alpha$ for some $\alpha \in \mathbb{R}^n$. Thus,

$$\boldsymbol{M}^\dagger\boldsymbol{M}\theta_t = \boldsymbol{M}^\dagger\boldsymbol{M}\boldsymbol{M}^\dagger\alpha = \boldsymbol{M}^\dagger\alpha = \theta_t\,. \tag{74}$$

This finishes the proof. □

**Lemma 2.** *$\mathcal{E}(\theta)$ is 1-smoothness.*

*Proof.* Recall the prediction error of $\theta \in \mathbb{R}^d$

$$\mathcal{E}(\theta) = \frac{1}{2}\|\Phi\theta - v\|^2_{\boldsymbol{D}_\mu} = \frac{1}{2}\|\theta - \theta^*\|^2_\Sigma\,, \tag{75}$$

where $\Sigma = \Phi^\top\boldsymbol{D}_\mu\Phi$. The gradient of $\theta$ is $\mathcal{E}'(\theta) = \Sigma(\theta - \theta^*)$. Then

$$\|\mathcal{E}'(\theta_1) - \mathcal{E}'(\theta_2)\| = \|\Sigma(\theta_1 - \theta_2)\| \leq \lambda_{\max}(\Sigma)\,\|(\theta_1 - \theta_2)\| \leq \|\theta_1 - \theta_2\|\,, \tag{76}$$

where we use $\|\phi(s)\| \leq 1$ for all $s \in \mathcal{S}$ and $\lambda_{\max}(\Sigma) \leq \sum_s \mu(s)\lambda_{\max}(\phi(s)\phi(s)^\top) \leq 1$. □

**Lemma 3.** *Let $\varepsilon_{app} = \boldsymbol{N}\Pi^\perp_{\boldsymbol{M}}\theta^*$ and $\sigma_{stat} = \boldsymbol{H}(P - \hat{P})\Phi\theta^*$. We have*

$$\boldsymbol{H}v = \boldsymbol{M}\theta^* = \sum_{i=0}^{T-1}(\gamma\boldsymbol{W})^i\,(\boldsymbol{R} + \gamma(\varepsilon + \sigma)) + (\gamma\boldsymbol{W})^T(\boldsymbol{M}\theta^*)\,.$$

*for any $T \geq 1$.*

*Proof.* A simple recursive argument shows that,

$$\begin{aligned}\boldsymbol{M}\theta^* &= \boldsymbol{R} + \gamma\boldsymbol{H}P\Phi\theta^*\\ &= \boldsymbol{R} + \gamma\boldsymbol{N}\theta^* + \gamma\boldsymbol{H}(P - \hat{P})\Phi\theta^*\\ &= \boldsymbol{R} + \gamma(\boldsymbol{W}\boldsymbol{M} + \boldsymbol{N}\Pi^\perp_{\boldsymbol{M}})\theta^* + \gamma\boldsymbol{H}(P - \hat{P})\Phi\theta^*\\ &= \boldsymbol{R} + \gamma(\varepsilon_{\text{app}} + \varepsilon_{\text{stat}}) + \gamma\boldsymbol{W}\boldsymbol{M}\theta^*\\ &= \ldots\\ &= \sum_{i=0}^{T-1}(\gamma\boldsymbol{W})^i\,(\boldsymbol{R} + \gamma(\varepsilon_{\text{app}} + \varepsilon_{\text{stat}})) + (\gamma\boldsymbol{W})^T(\boldsymbol{M}\theta^*)\,.\end{aligned}$$

□

*Proof.* By Theorem 3, $\theta_t = \boldsymbol{M}^\dagger \mathcal{T}^{t-1}(\boldsymbol{R} + \gamma \boldsymbol{N}\theta_0)$ is the output of FVI at iteration $t$. By noting that $\mathcal{E}(\theta^*) = 0$ and Lemma 2, for any $\theta \in \mathbb{R}^d$.

$$\mathcal{E}(\theta) \leq \frac{1}{2}\left\|\theta - \theta^*\right\|^2 = \frac{1}{2}\left(\|\theta - \theta^*\|^2_{\boldsymbol{M}^\dagger \boldsymbol{M}} + \|\theta - \theta^*\|^2_{\boldsymbol{I}_d - \boldsymbol{M}^\dagger \boldsymbol{M}}\right) . \tag{77}$$

We first consider the second term. By Lemma 1,

$$\|\theta_t - \theta^*\|^2_{\boldsymbol{I}_d - \boldsymbol{M}^\dagger \boldsymbol{M}} = (\theta_t - \theta^*)^\top \left(\boldsymbol{I}_d - \boldsymbol{M}^\dagger \boldsymbol{M}\right)(\theta_t - \theta^*) = \|\theta^*\|^2_{\boldsymbol{I}_d - \boldsymbol{M}^\dagger \boldsymbol{M}} . \tag{78}$$

We now consider the first term. By Lemma 3,

$$\boldsymbol{M}(\theta^* - \theta_t) = \boldsymbol{M}\theta^* - \mathcal{T}^{t-1}(\boldsymbol{R} + \gamma \boldsymbol{N}\theta_0) \tag{79}$$

$$= \sum_{i=0}^{t-2} (\gamma \boldsymbol{W})^i \left(\boldsymbol{R} + \gamma(\varepsilon_{\text{app}} + \varepsilon_{\text{stat}})\right) + (\gamma \boldsymbol{W})^{t-1}(\boldsymbol{M}\theta^*) - \sum_{i=0}^{t-2} (\gamma \boldsymbol{W})^i \boldsymbol{R} - (\gamma \boldsymbol{W})^{t-1}(\boldsymbol{R} + \gamma \boldsymbol{N}\theta_0) \tag{80}$$

$$= \gamma \left(\sum_{i=0}^{t-2} (\gamma \boldsymbol{W})^i \varepsilon_{\text{app}} + \sum_{i=0}^{t-1} (\gamma \boldsymbol{W})^i \varepsilon_{\text{stat}} + (\gamma \boldsymbol{W})^{t-1} \boldsymbol{N}(\theta^* - \theta)\right) \tag{81}$$

Let $\hat{\Sigma} = \boldsymbol{M}^\top \boldsymbol{D}_k \boldsymbol{M}$ be the empirical covariance matrix. Note that $\lambda_{\min}(\boldsymbol{M}^\top \boldsymbol{M})/k \geq \lambda_{\min}(\hat{\Sigma})$. To show this, let $\bar{\boldsymbol{D}} = \text{diag}(\frac{1}{k}, \ldots, \frac{1}{k})$, and $\boldsymbol{M} = \boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^\top$ be the SVD of $\boldsymbol{M}$. Then

$$\begin{aligned}
\lambda^+_{\min}(\boldsymbol{M}^\top \bar{\boldsymbol{D}} \boldsymbol{M}) &= \min_{\|x\|=1} x^\top \boldsymbol{M}^\top \bar{\boldsymbol{D}} \boldsymbol{M} x && (82)\\
&= \min_{\|\alpha\|=1} \alpha^\top \boldsymbol{S}\boldsymbol{U}^\top \bar{\boldsymbol{D}} \boldsymbol{U}\boldsymbol{S}\alpha && (83)\\
&\geq \min_{\|\alpha\|=1} \alpha^\top \boldsymbol{S}\boldsymbol{U}^\top \boldsymbol{D}_k \boldsymbol{U}\boldsymbol{S}\alpha && (84)\\
&= \lambda^+_{\min}(\boldsymbol{M}^\top \boldsymbol{D}_k \boldsymbol{M}) && (85)
\end{aligned}$$

where we replace $x = \boldsymbol{V}\alpha$ since $\boldsymbol{V}$ are orthonormal bases, and $\min_{i \in [k]} \hat{\mu}_i \leq 1/k$. Therefore,

$$\|\boldsymbol{M}^\dagger\| = 1/\sqrt{\lambda^+_{\min}(\boldsymbol{M}^\dagger \boldsymbol{M})} \leq 1/\sqrt{k\lambda^+_{\min}(\hat{\Sigma})}.$$

Combining the results above we have,

$$\begin{aligned}
\|\theta_t - \theta^*\|^2_{\boldsymbol{M}^\dagger \boldsymbol{M}} &= \left\|\boldsymbol{M}^\dagger \boldsymbol{M}(\theta_t - \theta^*)\right\|^2 && (86)\\
&\leq \|\boldsymbol{M}^\dagger\|^2 \left\|\boldsymbol{M}(\theta_t - \theta^*)\right\|^2 && (87)\\
&\leq \frac{\gamma}{k\lambda_{\min}(\hat{\Sigma})} \left\|\sum_{i=0}^{t-1} (\gamma \boldsymbol{W})^i (\varepsilon_{\text{app}} + \varepsilon_{\text{stat}}) + (\gamma \boldsymbol{W})^{t-1} \boldsymbol{N}(\theta^* - \theta)\right\|^2 && (88)\\
&\leq \frac{4\gamma}{k\lambda_{\min}(\hat{\Sigma})} \left((\varepsilon^2 + \sigma^2)\left\|\sum_{i=0}^{t-1} (\gamma \boldsymbol{W})^i\right\|^2 + \left\|(\gamma \boldsymbol{W})^{t-1}\right\|^2 \|\Phi\|^2 \|\theta_0 - \theta^*\|^2\right) . && (89)
\end{aligned}$$

Combine this with Eq. (78) finishes the proof.

□

## A.6 Proof of Corollary 2

*Proof.* Recall that in the proof of Theorem 5 we have

$$\|\theta_t - \theta^*\|^2_{\boldsymbol{M}^\dagger \boldsymbol{M}} \leq \frac{4\gamma}{k\lambda_{\min}(\hat{\Sigma})} \left((\varepsilon^2 + \sigma^2)\left\|\sum_{i=0}^{t-1} (\gamma \boldsymbol{W})^i\right\|^2 + \left\|(\gamma \boldsymbol{W})^{t-1}\right\|^2 \|\Phi\|^2 \|\theta_0 - \theta^*\|^2\right) . \tag{90}$$

Given that $\|\boldsymbol{W}\| < 1$, we have for the fixed point $\theta_\infty$,

$$\|\theta_t - \theta^*\|^2_{\boldsymbol{M}^\dagger \boldsymbol{M}} \le \frac{4\gamma(\varepsilon^2 + \sigma^2)}{k\lambda_{\min}(\hat{\Sigma})(1-\gamma)^2} \tag{91}$$

We first consider $\sigma^2 = \|\boldsymbol{H}(P - \hat{P})v\|^2$. By Hoeffding's inequality and a union bound we have with probability at least $1-\delta$, for any $s \in \operatorname{supp}(\boldsymbol{D})$,

$$\left|(\hat{P}_s - P_s)^\top v\right| \le \frac{1}{1-\gamma}\sqrt{\frac{\log(|\mathcal{S}|/\delta)}{2n(s)}}\,. \tag{92}$$

Thus, let $n_{\min} = \min_{s:n(s)>0} n(s)$, we have

$$\frac{\sigma^2}{k} \le \frac{\log(|\mathcal{S}|/\delta)}{2(1-\gamma)^2 n_{\min}}\,. \tag{93}$$

Now we consider $\varepsilon^2 = \|\boldsymbol{N}\Pi^\perp_{\boldsymbol{M}}\theta^*\|^2$. Since $\boldsymbol{N}\Pi^\perp_{\boldsymbol{M}}$ is perpendicular to $\boldsymbol{M}$, and all features have norm bounded by one,

$$\frac{\varepsilon^2}{k} \le \|\theta^*\|^2_{\boldsymbol{I}_d - \boldsymbol{M}^\dagger \boldsymbol{M}}\,. \tag{94}$$

Combine the above we have,

$$\begin{aligned}
\mathcal{E}(\theta) &\le \frac{1}{2}\|\theta - \theta^*\|^2_{\boldsymbol{M}^\dagger \boldsymbol{M}} + \frac{1}{2}\|\theta^*\|^2_{\boldsymbol{I}_d - \boldsymbol{M}^\dagger \boldsymbol{M}} && (95)\\
&\le \frac{2\gamma}{\lambda_{\min}(\hat{\Sigma})(1-\gamma)^2}\left(\frac{\log(|\mathcal{S}|/\delta)}{2(1-\gamma)^2 n_{\min}} + \|\theta^*\|^2_{\boldsymbol{I}_d - \boldsymbol{M}^\dagger \boldsymbol{M}}\right) + \frac{1}{2}\|\theta^*\|^2_{\boldsymbol{I}_d - \boldsymbol{M}^\dagger \boldsymbol{M}} && (96)\\
&= \frac{\gamma\log(|\mathcal{S}|/\delta)}{\lambda_{\min}(\hat{\Sigma})(1-\gamma)^4 n_{\min}} + \frac{4\gamma}{\lambda_{\min}(\hat{\Sigma})(1-\gamma)^2}\|\theta^*\|^2_{\boldsymbol{I}_d - \boldsymbol{M}^\dagger \boldsymbol{M}} && (97)
\end{aligned}$$

Finally, using the tower rule gives the desired result.

□

### A.7 Concentration of Eigenvalues and Bounding the Orthogonal Complement

We will need the following result (Kuzborskij et al., 2021, Theorem 6), which is concerned with the magnitude of projection onto the eigenspace of a covariance matrix. The result is based on (Shawe-Taylor et al., 2005, Theorem 1)

**Lemma 4.** *Let $\hat{\Sigma} = \frac{1}{n}\sum_i x_i x_i^\top$ be the covariance matrix of i.i.d. data $x_i \in \mathbb{R}^d$. Denote the $h$-"tail" of eigenvalues of a covariance matrix $\hat{\Sigma} =$ as*

$$\lambda^{>h} = \sum_{i=h+1}^{n} \lambda_i\,. \tag{98}$$

*Let $\boldsymbol{U}_r$ be the first $r$ eigenbasis for $r \in [n]$. Then for any $z \in \mathbb{R}^d$, with probability at least $1-\delta$,*

$$\mathbb{E}\left[\|\Pi^\perp_{\boldsymbol{U}_r} z\|_2^2\right] \le \min_{h\in[r]}\left\{\frac{1}{n}\lambda^{>h} + \frac{1+\sqrt{h}}{\sqrt{n}}\sqrt{\frac{2}{n}\sum_{i=1}^{n}\|x_i\|^2}\right\} + \|z\|_2^2\sqrt{\frac{18}{n}\ln\left(\frac{2n}{\delta}\right)}\,. \tag{99}$$

The next lemma gives a non-asymptotic result to understand the behaviour of $\hat{\lambda}_{\min}$ (Kuzborskij et al., 2021, Lemma 1).

**Lemma 5.** *Let $\boldsymbol{X} = [\boldsymbol{X}_1, \dots, \boldsymbol{X}_n] \in \mathbb{R}^{d\times n}$ be a random matrix with i.i.d. columns, such that $\max_i \|\boldsymbol{X}_i\|_2 \le K$, and let $\hat{\Sigma} = \boldsymbol{X}\boldsymbol{X}^\top/n$, and $\Sigma = \mathbb{E}[\boldsymbol{X}_1\boldsymbol{X}_1^\top]$. Then, for every $\alpha \ge 0$, with probability at least $1 - 2e^{-\alpha}$, we have*

$$\lambda^+_{\min}(\hat{\Sigma}) \ge \lambda^+_{\min}(\Sigma)\left(1 - K^2\left(c\sqrt{\frac{d}{n}} + \sqrt{\frac{\alpha}{n}}\right)\right)^2_+ \quad \textit{for } n \ge d\,, \tag{100}$$

*and furthermore, assuming that* $\|\boldsymbol{X}_i\|_{\Sigma^\dagger} = \sqrt{d}$*, for all* $i \in [n]$*, we have*

$$\lambda^+_{\min}(\hat{\Sigma}) \geq \lambda^+_{\min}(\Sigma) \left( \sqrt{\frac{d}{n}} - K^2 \left( c + 6\sqrt{\frac{\alpha}{n}} \right) \right)^2_+ \quad \textit{for } n < d \, , \tag{101}$$

*where we have an absolute constant* $c = 2^{3.5}\sqrt{\ln 9}$.

### A.8 Neural Tangent Kernel Extension

We show how to extend the results of Theorem 3 using neural tangent kernel. Let $v(s, \omega)$ be the value function approximated by a neural network, where $\omega$ is all the parameters of the neural network. The neural tangent kernel is defined as

$$\text{ker}(s, s') = \mathbb{E}_{\omega \sim \Omega} \left\langle \frac{\partial v(s, \omega)}{\partial \omega}, \frac{\partial v(s', \omega)}{\partial \omega} \right\rangle \tag{102}$$

where $s, s'$ are two states, $\Omega$ is an initialization distribution over $\omega$. Let $\omega_0$ be random initial parameters sampled from $\Omega$, there is

$$\text{ker}_0(s, s') = \left\langle \frac{\partial v(s, \omega_0)}{\partial \omega_0}, \frac{\partial v(s', \omega_0)}{\partial \omega_0} \right\rangle \approx \text{ker}(s, s') \, . \tag{103}$$

See (Arora et al., 2019, Theorem 3.1) for the proof. Note that in this kernel, an input $s$ is mapped to a feature vector $\phi(s) = \frac{\partial v(s, \omega_0)}{\partial \omega_0}$.

Consider a least square problem: given a training dataset $\{x_i, y_i\}_{i=1}^k$, consider training the neural network by minimizing the squared loss over training data: $\frac{1}{2}\sum_{i=1}^k (v(x_i, \omega) - y_i)^2$. The following result shows that a properly randomly initialized sufficiently wide deep neural network trained by gradient descent is equivalent to a kernel regression predictor with neural tangent kernel (Arora et al., 2019, Theorem 3.2),

$$v(x, \omega) \approx v_{\text{NTK}}(x) = \text{ker}(x, \boldsymbol{X})^\top \text{ker}(\boldsymbol{X}, \boldsymbol{X})^{-1} \mathbf{y} \, , \tag{104}$$

where $\text{ker}(x, \boldsymbol{X}) = (\text{ker}(x, x_1), \ldots, \text{ker}(x, x_k))^\top$, $\text{ker}(\boldsymbol{X}, \boldsymbol{X}) \in \mathbb{R}^{k \times k}$ whose $(i, j)$-th element is $\text{ker}(x, x_i)$, and $\mathbf{y} \in \mathbb{R}^k$ whose $i$-th element is $y_i$.

Recall that $\{x_i, r_i, x'_i\}_{i=1}^k$ is the batch dataset on the support of the $\hat{\mu}$. Let $\text{ker}(\boldsymbol{X}', \boldsymbol{X})$ be a $n \times n$ matrix where the $(i, j)$-th element is $\text{ker}(x'_i, x_j)$. For simplicity, we assume that FVI uses zero value prediction initially. We now prove that for any $s \in \mathcal{S}$, the value prediction at the $t$-th iteration of FVI is given by

$$v_t(s) = \text{ker}(s, \boldsymbol{X})^\top \text{ker}(\boldsymbol{X}, \boldsymbol{X})^{-1} \left( \sum_{i=1}^{t-1} \left( \gamma \text{ker}(\boldsymbol{X}', \boldsymbol{X}) \text{ker}(\boldsymbol{X}, \boldsymbol{X})^{-1} \right)^i \boldsymbol{R} \right) . \tag{105}$$

We prove this result by induction. For the base case $i = 1$, the result directly follows by Eq. (104),

$$v_1(s) = \text{ker}(s, \boldsymbol{X})^\top \text{ker}(\boldsymbol{X}, \boldsymbol{X})^{-1} \boldsymbol{R} . \tag{106}$$

Now suppose the induction hypothesis holds for $i = t - 1$, then by Eq. (104)

$$v_t(s) = \text{ker}(s, \boldsymbol{X})^\top \text{ker}(\boldsymbol{X}, \boldsymbol{X})^{-1} \tag{107}$$

$$\left( \boldsymbol{R} + \gamma \text{ker}(\boldsymbol{X}', \boldsymbol{X})^\top \text{ker}(\boldsymbol{X}, \boldsymbol{X})^{-1} \left( \sum_{i=1}^{t-2} \left( \gamma \text{ker}(\boldsymbol{X}', \boldsymbol{X}) \text{ker}(\boldsymbol{X}, \boldsymbol{X})^{-1} \right)^i \boldsymbol{R} \right) \right) \tag{108}$$

$$= \text{ker}(s, \boldsymbol{X})^\top \text{ker}(\boldsymbol{X}, \boldsymbol{X})^{-1} \left( \sum_{i=1}^{t-1} \left( \gamma \text{ker}(\boldsymbol{X}', \boldsymbol{X}) \text{ker}(\boldsymbol{X}, \boldsymbol{X})^{-1} \right)^i \boldsymbol{R} \right) . \tag{109}$$

Then, given that the following matrix is a contractor

$$\gamma \text{ker}(\boldsymbol{X}', \boldsymbol{X}) \text{ker}(\boldsymbol{X}, \boldsymbol{X})^{-1} \, , \tag{110}$$

we have that at the convergence, the value prediction of FVI for any $s \in \mathcal{S}$ is given by

$$v_{\text{FVI}}(s) = \text{ker}(s, \boldsymbol{X})^\top \text{ker}(\boldsymbol{X}, \boldsymbol{X})^{-1} \left( \boldsymbol{I} - \gamma \text{ker}(\boldsymbol{X}', \boldsymbol{X}) \text{ker}(\boldsymbol{X}, \boldsymbol{X})^{-1} \right) \boldsymbol{R} . \tag{111}$$

Note that ker$(\boldsymbol{X}', \boldsymbol{X})$ker$(\boldsymbol{X}, \boldsymbol{X})^{-1}$ corresponds to the core matrix $\boldsymbol{W}$ in the linear setting.

## B Experiment setup

In this section, we provide additional details about the experimental setup and hyper-parameters used for each of the environments. For all of these environments the regularization weights were considered as tunable hyper-parameters. In addition, for $R_\phi$ (see Eq 23), the scale factor $\beta$ was also considered as a parameter to be tuned in order to approximate the feature matrix norm.

### B.1 Acrobot

- Replay buffer with 10k tuples sampled using a random policy across trajectories with maximum episode length of 64.
- A DQN with hidden units consisting of fully connected layers with $(100, 100)$ units.
- Batch size 64.
- Learning rate of 1e-3.
- Regularized RM with weight of 2e-2 on $\mathcal{R}_\phi$ and 1e-4 on $\mathcal{R}_w$.
- Regularized TD with weight of 0 on $\mathcal{R}_\phi$ and 1e-4 on $\mathcal{R}_w$.

### B.2 Reacher

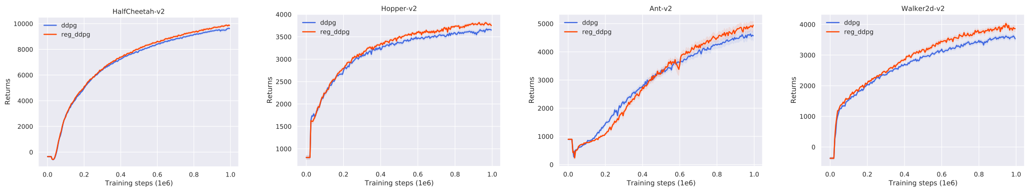
- Replay buffer with 10k tuples sampled from a random policy across trajectories with maximum steps per episode of length 50.
- Learning rate 1e-4.
- A value network for the continuous action inputs with a fc observation layer with params (50,), a fc action layer with params (50,) and a joint fc layer with params (100,).
- Batch size 64.
- Gradient clipping with a norm of 10.0
- Regularized RM with weight of 0.15 on $\mathcal{R}_w$ and 0 on $\mathcal{R}_\phi$.
- Regularized TD with weight of 2e-2 on $\mathcal{R}_w$ and 7e-3 on $\mathcal{R}_\phi$.

### B.3 Cartpole

- Replay buffer with 10k tuples sampled using a random policy across trajectories with maximum steps per episode of length 50.
- A DQN with hidden units consisting of fully connected layers with $(100, 100)$ units.
- Batch size 64.
- Learning rate 1e-3.
- Regularized RM with weight of 0.29 on $\mathcal{R}_w$ and 0 on $\mathcal{R}_\phi$.
- Regularized TD with weight of 1.5e-3 on $\mathcal{R}_w$ and 5e-3 on $\mathcal{R}_\phi$.

### B.4 Pendulum

- Replay buffer with 1k tuples obtained by sampling directly from a fixed initial state distribution using a random policy.
- A value network for the continuous action inputs with a fc observation layer with params (50,), a fc action layer with params (50,) and a joint fc layer with params (100,).
- Batch size 64.
- Learning rate 1e-3.
- Regularized RM with weight of 1.0 on $\mathcal{R}_w$ and 5.4e-4 on $\mathcal{R}_\phi$.
- Regularized TD with weight of 0 on $\mathcal{R}_w$ and 1.0 on $\mathcal{R}_\phi$.



(a) HalfCheetah (b) Hopper (c) Ant (d) Walker2d

Figure 3: We show the results with proposed regularization compared to the baseline algorithms. The algorithms are trained using a fixed offline data set collected by random initialized policies. The $x$-axis shows the training iterations (in thousands) and $y$-axis shows the performance. All plots are averaged over 100 runs. The shaded area shows the standard error.

### B.5 Extra Experiment Results

We provide extra experiment results on four mujoco control problems to assess the applicability of the proposed regularization $R_\phi$: HalfCheetah, Hopper, Ant, and Walker2d. The results are provided in Fig. 3. All results are averaged over 100 runs with different random seeds. The hyper-parameters are provided below.

- The Q-function is approximated by two hidden layer fully neural networks, where the hidden layer size is 256.
- Batch size 256.
- Learning rate 3e-4.
- Regularized TD weight are tuned from $\{1e-4, 1e-3, 1e-2, 1e-1, 1\}$