An MRP Formulation for Supervised Learning: Generalized Temporal Difference Learning Models

Yangchen Pan[*]
Department of Engineering Science
University of Oxford
Oxford, United Kingdom
yangchen.pan@eng.ox.ac.uk

Junfeng Wen[*]
School of Computer Science
Carleton University
Ottawa, Canada
junfengwen@gmail.com

Chenjun Xiao
Department of Computing Science
University of Alberta
Edmonton, Canada
chenjun@ualberta.ca

Philip H.S. Torr
Department of Engineering Science
University of Oxford
Oxford, United Kingdom
philip.torr@eng.ox.ac.uk

Abstract

In traditional statistical learning, data points are usually assumed to be independently and identically distributed (i.i.d.) following an unknown probability distribution. This paper presents a contrasting viewpoint, perceiving data points as interconnected and employing a Markov reward process (MRP) for data modeling. We reformulate the typical supervised learning as an on-policy policy evaluation problem within reinforcement learning (RL), introducing a generalized temporal difference (TD) learning algorithm as a resolution. Theoretically, our analysis draws connections between the solutions of linear TD learning and ordinary least squares (OLS). We also show that under specific conditions, particularly when noises are correlated, the TD's solution proves to be a more effective estimator than OLS. Furthermore, we establish the convergence of our generalized TD algorithms under linear function approximation. Empirical studies verify our theoretical results, examine the vital design of our TD algorithm and show practical utility across tasks such as regression and image classification with deep learning.

[*] Equal contribution. Correspondence to Yangchen Pan and Junfeng Wen.

38th Workshop on Aligning Reinforcement Learning Experimentalists and Theorists (ARLET 2024).

1 Introduction

The primary objective of statistical supervised learning (SL) is to learn the relationship between the features and the output (response) variable. To achieve this, generalized linear models (Nelder & Wedderburn, 1972; McCullagh & Nelder, 1989) are considered a generic algorithmic framework employed to derive objective functions. These models make specific assumptions regarding the conditional distribution of the response variable given input features, which can take forms such as Gaussian (resulting in ordinary least squares), Poisson (resulting in Poisson regression), or multinomial (resulting in logistic regression or multiclass softmax cross-entropy loss).

In recent years, reinforcement learning (RL), widely utilized in interactive learning settings, has witnessed a surge in popularity. This surge has attracted growing synergy between RL and SL, where each approach complements the other in various ways. In SL-assisted RL, the area of imitation learning (Hussein et al., 2017) may leverage expert data to regularize/speed up RL, while weakly supervised methods (Lee et al., 2020) have been adopted to constrain RL task spaces, and relabeling techniques contributed to goal-oriented policy learning (Ghosh et al., 2021). Conversely, RL has also expanded its application into traditional SL domains. RL has proven effective in fine-tuning large language models (MacGlashan et al., 2017), aligning them with user preferences. Additionally, RL algorithms (Gupta et al., 2021) have been tailored for training neural networks (NNs), treating individual network nodes as RL agents. In the realm of imbalanced classification, RL-based control algorithms have been developed, where predictions correspond to actions, rewards are based on heuristic correctness criteria, and episodes conclude upon incorrect predictions within minority classes (Lin et al., 2020). Permative prediction (Perdomo et al., 2020) provides a theoretical framework that can deal with nonstationary data and it can be reframed within a RL context.

Existing approaches that propose a RL framework for solving SL problems often exhibit a heuristic nature. These approaches involve crafting specific formulations, including elements like agents, reward functions, action spaces, and termination conditions, based on intuitive reasoning tailored to particular tasks. Consequently, they lack generality, and their heuristic nature leaves theoretical assumptions, connections between optimal RL and SL solutions, and convergence properties unclear. To the best of our knowledge, it remains uncertain whether a unified and systematic RL formulation capable of modeling a wide range of conventional SL problems exists. Such a formulation should be agnostic to learning settings, including various tasks such as ordinary least squares regression, Poisson regression, binary or multi-class classification, etc.

In this study, we introduce a generic Markov process formulation for data generation, offering an alternative to the conventional i.i.d. data assumption in SL. Specifically, when faced with a SL dataset, we view the data points as originating from a Markov reward process (MRP) (Szepesvari, 2010). To accommodate a wide range of problems, such as Poisson regression, binary or multi-class classification, we introduce a generalized TD learning model in Section 3. Section 4 explores the relationship between the solutions obtained through TD learning and the original linear regression. Furthermore, we prove that under specific conditions with correlated noise, TD estimator is more efficient than the traditional ordinary least squares (OLS) estimator. We provide convergence result in Section 5 under linear function approximation. Our paper concludes with an empirical evaluation of our TD algorithm in Section 6, verifying our theoretical results, assessing its critical design choices and practical utility when integrated with a deep neural network across various tasks, achieving competitive results and, in some cases, improvements in generalization performance. We view our work as a step towards unifying diverse learning tasks from two pivotal domains within a single, coherent theoretical framework.

2 Background

This section provides a brief overview of the relevant concepts from statistical SL and RL settings.

2.1 Conventional Supervised Learning

In the context of statistical learning, we make the assumption that data points, in the form of $(\mathbf{x}, y) \in \mathcal{X} \times \mathcal{Y}$, are independently and identically distributed (i.i.d.) according to some unknown probability distribution P. The goal is to find the relationship between the feature $\mathbf{x}$ and response variable y given a training dataset $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$.

In a simple linear function approximation case, a commonly seen algorithm is ordinary least squares (OLS) that optimizes squared error objective function

$$\min_{\mathbf{w}} ||\mathbf{X}\mathbf{w} - \mathbf{y}||_2^2. \tag{1}$$

where $\mathbf{X}$ is the $n \times d$ feature matrix and $\mathbf{y}$ is the corresponding n-dimensional training label vector, and the $\mathbf{w}$ is the parameter vector we aim to optimize. From a probabilistic perspective, this objective function can be derived by assuming $p(y|\mathbf{x})$ follows a Gaussian distribution with mean $\mathbf{x}^\top \mathbf{w}$ and conducting maximum likelihood estimation (MLE) for $\mathbf{w}$ with the training dataset. It is well known that $\mathbb{E}[Y|\mathbf{x}]$ is the optimal predictor (Bishop, 2006). For many other choices of distribution $p(y|\mathbf{x})$, generalized linear models (GLMs) (Nelder & Wedderburn, 1972) are commonly employed for estimating $\mathbb{E}[Y|\mathbf{x}]$. This includes OLS, Poisson regression (Nelder, 1974) and logistic regression, etc.

An important concept in GLMs is the *inverse link function*, which we denoted as f, that establishes a connection between the linear prediction (also called the **logit**), and the conditional expectation: $\mathbb{E}[Y|\mathbf{x}] = f(\mathbf{x}^\top \mathbf{w})$. For example, in logistic regression, the inverse link function is the sigmoid function. We later propose generalized TD learning models within the framework of RL that correspond to GLMs, enabling us to handle a wide range of data that are modled by different distributions in supervised learning.

2.2 Reinforcement Learning

Reinforcement learning is often formulated within the Markov decision process (MDP) framework. An MDP can be represented as a tuple $(\mathcal{S}, \mathcal{A}, r, P, \gamma)$ (Puterman, 2014), where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, $r : \mathcal{S} \times \mathcal{A} \mapsto \mathbb{R}$ is the reward function, $P(\cdot|s, a)$ defines the transition probability, and $\gamma \in (0, 1]$ is the discount factor. Given a policy $\pi : \mathcal{S} \times \mathcal{A} \to [0, 1]$, the return at time step t is $G_t = \sum_{i=0}^{\infty} \gamma^i r(S_{t+i}, A_{t+i})$, and value of a state $s \in \mathcal{S}$ is the expected return starting from that state $v^\pi(s) = \mathbb{E}_\pi[G_t|S_t = s]$. In this work, we focus on the policy evaluation problem for a fixed policy, thus the MDP can be reduced to a Markov reward process (MRP) (Szepesvari, 2010) described by $(\mathcal{S}, r^\pi, P^\pi, \gamma)$ where $r^\pi(s) \overset{\text{def}}{=} \sum_a \pi(a|s) r(s, a)$ and $P^\pi(s'|s) \overset{\text{def}}{=} \sum_a \pi(a|s) P(s'|s, a)$. When it is clear from the context, we will slightly abuse notations and ignore the superscript π.

In policy evaluation problem, the objective is to estimate the state value function of a fixed policy π by using the trajectory $s_0, r_1, s_1, r_2, s_2, ...$ generated from π. Under linear function approximation, the value function is approximated by a parametrized function $v(s) \approx \mathbf{x}(s)^\top \mathbf{w}$ with parameters $\mathbf{w}$ and some fixed feature mapping $\mathbf{x} : \mathcal{S} \mapsto \mathbb{R}^d$ where d is the feature dimension. Note that the state value satisfies the Bellman equation

$$v(s) = r(s) + \gamma \mathbb{E}_{S' \sim P(\cdot|s)}[v(S')]. \tag{2}$$

One fundamental approach for the evaluation problem is the temporal difference (TD) learning (Sutton, 1988), which uses a sampled transition s_t, r_{t+1}, s_{t+1} to update the parameters $\mathbf{w}$ through stochastic fixed point iteration based on (2) with a step-size $\alpha > 0$:

$$\mathbf{w} \leftarrow \mathbf{w} + \alpha(y_{t,td} - \mathbf{x}(s_t)^\top \mathbf{w})\mathbf{x}(s_t), \tag{3}$$

where $y_{t,td} \overset{\text{def}}{=} r_{t+1} + \gamma \mathbf{x}(s_{t+1})^\top \mathbf{w}$. To simplify notations and align concepts, we will use $\mathbf{x}_t \overset{\text{def}}{=} \mathbf{x}(s_t)$. In linear function approximation setting, TD converges to the solution that solves the system $\mathbf{A}\mathbf{w} = \mathbf{b}$ (Bradtke & Barto, 1996; Tsitsiklis & Van Roy, 1997), where

$$\mathbf{A} = \mathbb{E}[\mathbf{x}_t(\mathbf{x}_t - \gamma \mathbf{x}_{t+1})^\top] = \mathbf{X}^\top \mathbf{D}(\mathbf{I} - \gamma \mathbf{P})\mathbf{X}, \tag{4}$$

$$\mathbf{b} = \mathbb{E}[r_{t+1}\mathbf{x}_t] = \mathbf{X}^\top \mathbf{D}\mathbf{r} \tag{5}$$

with $\mathbf{X} \in \mathbb{R}^{|\mathcal{S}| \times d}$ being the feature matrix whose rows are the state features $\mathbf{x}_t$, $\mathbf{D} \in \mathbb{R}^{|\mathcal{S}| \times |\mathcal{S}|}$ being the diagonal matrix with the stationary distribution $D(s_t)$ on the diagonal, $\mathbf{P} \in \mathbb{R}^{|\mathcal{S}| \times |\mathcal{S}|}$ being the transition probability matrix (i.e., $\mathbf{P}_{ij} = P(s_j|s_i)$) and $\mathbf{r} \in \mathbb{R}^{|\mathcal{S}|}$ being the reward vector. Note that the matrix $\mathbf{A}$ is often invertible under mild conditions (Tsitsiklis & Van Roy, 1997).

3 MRP View and Generalized TD Learning

This section describes our MRP construction given the same dataset $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$ and proposes our generalized TD learning algorithm to solve it. This approach is based on the belief that these data points originate from some MRP, rather than being i.i.d. generated.

SL Definitions	RL Definitions
Feature matrix $\mathbf{X}$	Feature matrix $\mathbf{X}$
Feature of the ith example $\mathbf{x}_i$	The state feature $\mathbf{x}(s_i) = \mathbf{x}_i$
Training target y_i of $\mathbf{x}_i$	The state value $v(s_i) = y_i$

Table 1: How definitions in SL correspond to those in RL

Regression. We start by considering the basic regression setting with linear models before introducing our generalized TD algorithm. Table 1 summarizes how we can view concepts in the conventional SL from an RL perspective. The key is to treat the original training label as a state value that we are trying to learn, and then the reward function can be derived from the Bellman equation (2) as

$$r(s) = v(s) - \gamma \mathbb{E}_{S' \sim P(\cdot|s)}[v(S')]. \tag{6}$$

We will discuss the choice of $\mathbf{P}$ later. At each iteration (or time step in RL), the reward can be approximated using a stochastic example. For instance, assume that at iteration t (i.e., time step in RL), we obtain an example $(\mathbf{x}_i^{(t)}, y_i^{(t)})$. We use superscripts and subscripts to denote that the ith training example is sampled at the tth time step. Then the next example $(\mathbf{x}_j^{(t+1)}, y_j^{(t+1)})$ is sampled according to $P(\cdot|\mathbf{x}_i^{(t)})$ and the reward can be estimated as $r^{(t+1)} = y_i^{(t)} - \gamma y_j^{(t+1)}$ by approximating the expectation in Equation (6) with a stochastic example. As one might notice that, in a sequential setting the t is monotonically increasing, hence we will simply use a simplified notation $(\mathbf{x}_t, y_t)$ to denote the training example sampled at time step t.

We now summarize and compare the updating rules in conventional SL and in our TD algorithm under linear function approximation. At time step t, the **conventional updating rule** based on stochastic gradient descent (SGD) is

$$\mathbf{w} \leftarrow \mathbf{w} + \alpha(y_t - \mathbf{x}_t^\top \mathbf{w})\mathbf{x}_t, \tag{7}$$

while our **TD updating rule** is

$$\mathbf{w} \leftarrow \mathbf{w} + \alpha(y_{t,td} - \mathbf{x}_t^\top \mathbf{w})\mathbf{x}_t \tag{8}$$

$$\text{where } \; y_{t,td} \overset{\text{def}}{=} r_{t+1} + \gamma \widehat{y}_{t+1} = y_t - \gamma y_{t+1} + \gamma \mathbf{x}_{t+1}^\top \mathbf{w} \tag{9}$$

and $\mathbf{x}_{t+1} \sim P(\cdot|\mathbf{x}_t)$ with ground-truth label y_{t+1}. The critical difference is that TD uses a bootstrap, so it does not cancel the γy_{t+1} term from the reward when computing the TD training target $y_{t,td}$. By setting $\gamma = 0$, one recovers the original supervised learning updating rule (7).

Generalized TD: An extension to general learning tasks. A natural question regarding TD is how to extend it to different types of data, such as those with counting, binary, or multiclass labels. Recall that in generalized linear models (GLMs), it is assumed that the output variable $y \in \mathcal{Y}$ follows an exponential family distribution. In addition, there exists an *inverse link function* f that maps a linear prediction $z \overset{\text{def}}{=} \mathbf{w}^\top \mathbf{x}$ to the output/label space $\mathcal{Y}$ (i.e., $f(z) \in \mathcal{Y}, \forall z \in \mathbb{R}$). Examples of GLMs include linear regression (where y follows a Gaussian distribution, f is the identity function and the loss is the squared loss) and logistic regression (where y is Bernoulli, f is the sigmoid function and the loss is the log loss). More generally, the output may be in higher dimensional space and both z and y will be vectors instead of scalars. As an example, multinomial regression uses the softmax function f to convert a vector $\mathbf{z}$ to another vector $\mathbf{y}$ in the probability simplex. Interested readers can refer to Banerjee et al. (2005); Helmbold et al. (1999); McCullagh & Nelder (1989, Table 2.1) for more details. As per convention, we refer to z as **logit**.

Back to TD algorithm, the significance of the logit z is that it is naturally *additive*, which mirrors the additive nature of returns (cumulative sum of rewards) in RL. It also implies that one can add two linear predictions and the resultant $z = z_1 + z_2$ can still be transformed to a valid output $f(z) \in \mathcal{Y}$. In contrast, adding two labels does not necessarily produce a valid label $y_1 + y_2 \notin \mathcal{Y}$. Therefore, the idea is to construct a bootstrapped target in the real line (logit space, or z-space)

$$z_{t,td} \overset{\text{def}}{=} r_{t+1} + \gamma \widehat{z}_{t+1} = (z_t - \gamma z_{t+1}) + \gamma \mathbf{x}_{t+1}^\top \mathbf{w}$$

and then convert it back to the original label space to get the TD target $y_{t,td} = f(z_{t,td})$. In multiclass classification problems, we often use a one-hot vector to represent the original training target. For instance, in the case of MNIST, the target is a ten-dimensional one-hot vector. Consequently, the reward becomes a vector, with each component corresponding to a class. This can be interpreted as evaluating the policy under ten different reward functions in parallel and selecting the highest value for prediction.

Algorithm 1 provides the pseudo-code of our algorithm when using linear models. At time step t, the process begins by sampling the state $\mathbf{x}_t$, and then we sample the next state according to the

Dim. Trans. P	70	90	110	130
Random	0.027	0.075	$\leq 10^{-10}$	$\leq 10^{-10}$
Uniform	0.026	0.074	$\leq 10^{-10}$	$\leq 10^{-10}$
Distance (Far)	0.028	0.075	$\leq 10^{-10}$	$\leq 10^{-10}$
Distance (Close)	0.182	0.249	$\leq 10^{-10}$	$\leq 10^{-10}$
Deficient	0.035	0.172	0.782	0.650

Table 2: Distance between closed-form min-norm solutions of TD and OLS $\|\mathbf{w}_{TD} - \mathbf{w}_{LS}\|_2$. Input matrix $\mathbf{X}$ has normally distributed features with $n = 100$ and various dimensions d. Results are average over 10 runs. All standard errors are ≤ 0.02 except the deficient case ≤ 0.25. Details can be found in Appendix B.1.

predefined P. The reward is computed as the difference in logits after converting the original labels y_t, y_{t+1} into the logit space with the link function. Subsequently TD bootstrap target is constructed in the logit space. Finally, the TD target is transformed back to the original label space before it is used to calculate the loss. Note that in standard regression is a special case where the (inverse) link function is simply the identity function, so it reduces to the standard update (7) with squared loss. In practice, we might need some smoothing parameter when the function f^{-1} goes to infinity. For example, in binary classification, $\mathcal{Y} = \{0, 1\}$ and the corresponding logits are $z = -\infty$ and $z = \infty$. To avoid this, we subtract/add some small value to the label before applying f^{-1}.

Algorithm 1 Generalized TD for SL

```
Input: A dataset $\mathcal{D}$; randomly sample a data point $(\mathbf{x}_t, y_t) \in \mathcal{D}$ as the starting point. (One can also use mini-batch starting points in NNs.)
for $t = 1, 2, \dots$ do
    Sample $\mathbf{x}_{t+1} \sim P(\cdot|\mathbf{x}_t)$, let $y_{t+1}$ be its label
    $r_{t+1} = f^{-1}(y_t) - \gamma f^{-1}(y_{t+1})$ // $f^{-1}$ converts label to logits
    $z_{t,td} = r_{t+1} + \gamma \mathbf{x}_{t+1}^\top \mathbf{w}$ // Bootstrap target, a separate target network is needed in DNNs
    $\mathbf{w} \leftarrow \mathbf{w} - \alpha \nabla l(f(\mathbf{x}_t^\top \mathbf{w}), f(z_{t,td}))$ // $f$ converts logits back to label
```

4 MRP v.s. i.i.d. for Supervised Learning

As our MRP formulation is fundamentally different from the traditional i.i.d. view, this section discusses the property of TD's solution and its potential benefits in the linear setting and beyond. The section concludes with a discussion on the merits of adopting an MRP versus an i.i.d. perspective.

4.1 Connections between TD and OLS

The following proposition characterizes the connections between TD and OLS in the linear setting. Proof is in Appendix A.

Proposition 4.1. *[Connection between TD and OLS] When $\mathbf{D}$ has full support and $\mathbf{X}$ has linearly independent rows, TD and OLS have the same minimum norm solution. Moreover, any solution to the linear system $\mathbf{X}\mathbf{w} = \mathbf{y}$ must be also an solution to TD's linear system $\mathbf{A}\mathbf{w} = \mathbf{b}$ as defined in Equation* (5).

Empirical verification. Table 2 illustrates the distance between the closed-form solutions of our linear TD and OLS under various choices of the transition matrix by using a synthetic dataset (details are in Appendix B.1). Two key observations emerge: 1) As the feature dimension increases towards the overparameterization regime, both solutions become nearly indistinguishable, implying that designing $\mathbf{P}$ may be straightforward when employing a powerful model like NN. 2) Deficient choices for $\mathbf{P}$ with non-full support can pose issues and should be avoided. In practice, one might opt for a computationally and memory-efficient $\mathbf{P}$, where every entry is set to $1/n$. Such a matrix is ergodic and, therefore, not deficient. We will delve deeper into the selection of $\mathbf{P}$ below.

4.2 Statistical Efficiency and Variance Analysis

A natural question is under what condition the TD solution is better than the OLS solution, especially given that they may find the same solution under conditions specified above. Although the OLS estimator is known to be the best linear unbiased estimator (BLUE) under the i.i.d. assumption, our TD algorithm demonstrates the potential for a lower variance in settings with correlated noise, even when using a simple uniform transition matrix. Recall that conventional linear models assume $y_i = \mathbf{x}_i^\top \mathbf{w}^* + \epsilon_i$, where $\mathbf{w}^*$ is the true parameters and ϵ_i is assumed to be independent noise. In contrast, under the MRP perspective, we consider the possibility of correlated noise. The following proposition shows when a TD estimator will be the most efficient one:

Proposition 4.2. *Suppose* $\mathbf{A}$ *as in Equation* (5) *is invertible and the error vector* $\boldsymbol{\epsilon}$ *satisfies* $\mathbb{E}[\boldsymbol{\epsilon}|\mathbf{X}] = \mathbf{0}$ *and* $\text{Cov}[\boldsymbol{\epsilon}|\mathbf{X}] = \mathbf{C}$*. Then* $\mathbb{E}[\mathbf{w}_{TD}|\mathbf{X}] = \mathbf{w}^*$ *and the conditional covariance is*

$$\text{Cov}[\mathbf{w}_{TD}|\mathbf{X}] = \mathbf{A}^{-1}\mathbf{X}^\top \mathbf{S}\mathbf{C}\mathbf{S}^\top \mathbf{X}(\mathbf{A}^{-1})^\top \tag{10}$$

where $\mathbf{S} \overset{def}{=} \mathbf{D}(\mathbf{I} - \gamma\mathbf{P})$*. Moreover, if* $\mathbf{S} = \mathbf{S}^\top$*, the TD estimator is the BLUE for problems with* $\mathbf{C} = c \cdot \mathbf{S}^{-1}, \forall c > 0$.

Remark 1. This proposition identifies a situation in which our TD estimator outperforms other estimators, OLS included, in terms of efficiency. The condition $\mathbf{S} = \mathbf{S}^\top$ is needed so that there exists a corresponding symmetric covariance matrix $\mathbf{C}$ for the data. This condition implies that $\mathbf{DP} = \mathbf{P}^\top\mathbf{D}$, or $D(s_i)P(s_j|s_i) = D(s_j)P(s_i|s_j)$, which means the Markov chain is reversible (i.e., detailed balance). Also note that $\mathbf{S}, \mathbf{A}$ are invertible under mild conditions (e.g., ergodic Markov chain). In such cases, the TD estimator also corresponds to the generalized least squares (GLS) estimator (Aitken, 1936; Kariya & Kurata, 2004). In practice, one may be tempted to directly estimate the covariance matrix as done by feasible GLS (FGLS) (Baltagi, 2008). However, estimating a covariance matrix is nontrivial as demonstrated in Section 6. Furthermore, it should be emphasized that GLS/FGLS methods do not naturally support incremental learning, nor is it readily adaptable to deep learning models.

Remark 2. The benefits of TD may not be limited to the situations described by the above proposition. It is challenging to mathematically describe these benefits because, under more general $\mathbf{S}$, there is no intuitively interpretable form of the variance of TD's solution.

Below, we provide a more general perspective to understand the benefits of TD in terms of variance reduction. The basic idea is that when the ground truth target variables of consecutive time steps are positively correlated, the TD target benefits from a reduction in variance.

Proposition 4.3. *[Variance] Assume the estimated next-state value* $\widehat{y}_{t+1}$ *satisfies* $\widehat{y}_{t+1} = \mathbb{E}[y_{t+1}] + \epsilon$ *where* ϵ *is some independent noise with zero mean and standard deviation* σ_ϵ*. Let* σ_t *be the standard deviation of* y_t *and* $\rho_{t,t+1}$ *be the Pearson correlation coefficient between* y_t *and* y_{t+1}*. If* $\rho_{t,t+1} \geq \frac{\gamma^2\sigma_{t+1}^2 + \sigma_\epsilon^2}{2\gamma\sigma_t\sigma_{t+1}}$*, then* $\text{Var}(y_{t,td}) \leq \text{Var}(y_t)$.

Remark. To better interpret the result, we can consider $\sigma_t = \sigma_{t+1}$, then the variance of TD's target is simplified to $\text{Var}(y_{t,td}) = \sigma_t^2 + \gamma^2\sigma_t^2 - 2\gamma\rho_{t,t+1}\sigma_t^2 + \sigma_\epsilon^2$, which achieves its lowest value, $(1 - \rho_{t,t+1}^2)\sigma_t^2 + \sigma_\epsilon^2$ when $\gamma = \rho_{t,t+1}$. This suggests that the stronger correlation it is, the more (i.e. a larger γ) we might rely on the bootstrap term to reduce variance, coinciding with our intuition.

Empirical verification. Here we verify that when the outputs are indeed positively correlated, our method can generalize better than OLS. To this end, we run experiments using a Gaussian process with positive correlated outputs.

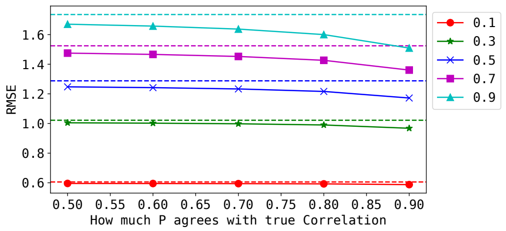


Figure 1: Comparing TD (solid) with OLS (dashed) with different settings.

For our TD method, we design the probability matrix as an interpolation between a covariance matrix $\mathbf{C}$ with some positive correlation and $\mathbf{1} - \mathbf{C}$, defined as $\mathbf{P} = (1 - \eta)(\mathbf{1} - \mathbf{C}) + \eta\mathbf{C}$, followed by normalization to ensure it forms a valid stochastic matrix. As η getting closer to one, the more our transition matrix agrees with the covariance matrix so that it is more likely to transition to positively correlated points. The results are shown in Figure 1. Under different levels of correlation, as indicated by various colors, TD increasingly outperforms OLS when the transition probability matrix $\mathbf{P}$ more closely aligns with the covariance.

The theoretical and empirical results from this section suggest: 1) in the absence of prior knowledge, adopting TD with small γ values and a uniform $\mathbf{P}$ could be beneficial for computational and memory efficiency; if the ground truth suggests that the data points are positively correlated, this approach might yield a performance gain; 2) when it is known that two points are positively correlated, one might enhance variance reduction by strategically encouraging transition from one point to another.

5 Convergence Analysis

In this section, we present convergence results for our generalized TD algorithm (Algorithm 1) under both the expected updating rule and the sample-based updating rule. Detailed proofs along with assumptions are provided in Appendix A.4. Here we show the finite-time convergence when using our TD(0) updates with linear function approximation. We primarily follow the convergence framework presented in Bhandari et al. (2018), making nontrivial adaptations due to the presence of the inverse link function. Let $z(s) = \mathbf{x}(s)^\top \mathbf{w}$, or $z = \mathbf{x}^\top \mathbf{w}$ for conciseness.

Convergence under expected update. The expected update rule is given by

$$\mathbf{w}_{t+1} = \mathbf{w}_t + \alpha \bar{g}(\mathbf{w}_t) \quad \text{with} \quad \bar{g}(\mathbf{w}_t) \overset{\text{def}}{=} \mathbb{E}[(y_{td} - y)\mathbf{x}] \tag{11}$$

where $y \overset{\text{def}}{=} f(z), y_{td} \overset{\text{def}}{=} f(r + \gamma \mathbf{x}'^\top \mathbf{w}_t)$ and $\mathbf{x}' \overset{\text{def}}{=} \mathbf{x}(s')$.

One can expand the distance from $\mathbf{w}_{t+1}$ to $\mathbf{w}^*$ as

$$\|\mathbf{w}^* - \mathbf{w}_{t+1}\|_2^2 = \|\mathbf{w}^* - \mathbf{w}_t\|_2^2 - 2\alpha(\mathbf{w}^* - \mathbf{w}_t)^\top \bar{g}(\mathbf{w}_t) + \alpha^2 \|\bar{g}(\mathbf{w}_t)\|_2^2$$

The common strategy is to make sure that the second term can outweigh the third term on the RHS so that $\mathbf{w}_{t+1}$ can get closer to $\mathbf{w}^*$ than $\mathbf{w}_t$ in each iteration. This can be achieved by choosing an appropriate step size as shown below:

Theorem 5.1. *[Convergence with Expected Update] Under Assumption A.1-A.4, consider the sequence* $(\mathbf{w}_0, \mathbf{w}_1, \cdots)$ *satisfying Equation* (11). *Let* $\overline{\mathbf{w}}_T \overset{\text{def}}{=} \frac{1}{T}\sum_{t=0}^{T-1} \mathbf{w}_t$, $\bar{z}_T = \mathbf{x}^\top \overline{\mathbf{w}}_T$ *and* $z^* = \mathbf{x}^\top \mathbf{w}^*$. *By choosing* $\alpha = \frac{1-\gamma L^2}{4L^3} > 0$, *we have*

$$\mathbb{E}\left[(z^* - \bar{z}_T)^2\right] \leq \left(\frac{2L^2}{1-\gamma L^2}\right)^2 \frac{\|\mathbf{w}^* - \mathbf{w}_0\|_2^2}{T} \tag{12}$$

$$\|\mathbf{w}^* - \mathbf{w}_T\|_2^2 \leq \exp\left(-T\omega \left(\frac{1-\gamma L^2}{2L^2}\right)^2\right) \|\mathbf{w}^* - \mathbf{w}_0\|_2^2 \tag{13}$$

Equation (12) shows that in expectation, the average prediction converges to the true value in the z-space, while Equation (13) shows that the last iterate converges to the fixed point exponentially fast when using expected update. In practice, we prefer sample-based updates is preferred and we discuss its convergence next.

Convergence under sample-based update. Below theorem shows the convergence under i.i.d. sample setting. Suppose s_t is sampled from the stationary distribution $D(s)$ and $s_{t+1} \sim P(\cdot|s_t)$. For conciseness, let $\mathbf{x}_t \overset{\text{def}}{=} \mathbf{x}(s_t)$ and $\mathbf{x}_{t+1} \overset{\text{def}}{=} \mathbf{x}(s_{t+1})$. Then the sample-based update rule is

$$\mathbf{w}_{t+1} = \mathbf{w}_t + \alpha_t g_t(\mathbf{w}_t) \text{ with } g_t(\mathbf{w}_t) \overset{\text{def}}{=} (y_{t,td} - y_t)\mathbf{x}_t \tag{14}$$

where $y_t \overset{\text{def}}{=} f(\mathbf{x}_t^\top \mathbf{w}_t), y_{t,td} \overset{\text{def}}{=} f(z_{t,td}) = f(r_{t+1} + \gamma \mathbf{x}_{t+1}^\top \mathbf{w}_t)$.

The following theorem shows the convergence when using i.i.d. sample for the update:

Theorem 5.2. *[Convergence with Sampled-based Update] Under Assumption A.1-A.4, with sample-based update Equation* (14), *let* $\sigma^2 = \mathbb{E}[\|g_t(\mathbf{w}^*)\|_2^2]$, $\overline{\mathbf{w}}_T \overset{\text{def}}{=} \frac{1}{T}\sum_{t=0}^{T-1} \mathbf{w}_t$, $\bar{z}_T = \mathbf{x}^\top \overline{\mathbf{w}}_T$ *and* $z^* = \mathbf{x}^\top \mathbf{w}^*$. *For* $T \geq \frac{64L^6}{(1-\gamma L^2)^2}$ *and a constant step size* $\alpha_t = 1/\sqrt{T}, \forall t$, *we have*

$$\mathbb{E}\left[(z^* - \bar{z}_T)^2\right] \leq \frac{L\left(\|\mathbf{w}^* - \mathbf{w}_0\|_2^2 + 2\sigma^2\right)}{\sqrt{T}(1-\gamma L^2)}. \tag{15}$$

This shows that the generalized TD update converges even when using sample-based update. Not surprisingly, it is slower than using the expected update (12).

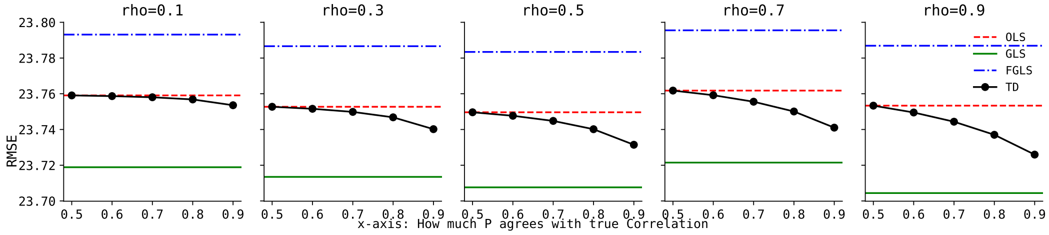


Figure 2: Test root mean squared error (RMSE) versus different Ps on exec time dataset. Each plot's x-axis represents the degree of alignment between the transition matrix P and the true covariance matrix that generates the noise. As P aligns better – implying a higher likelihood of data points with positively correlated noise transitioning from one to another – the solution of TD increasingly approximates the $\mathbf{w}^*$. Furthermore, as the correlation among the data intensifies, one can see larger gap between TD and OLS/FGLS.

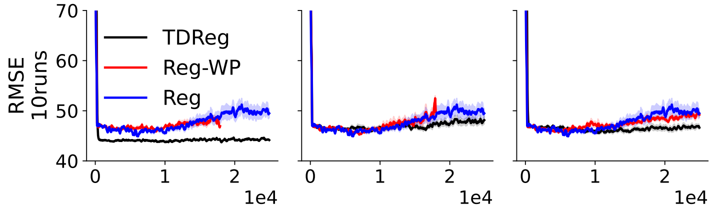


Figure 3: We show the test RMSE with error vs. the number of training steps when using three types of transition matrices. From left to right, the first type of transition assigns a high probability of transitioning to similar data points, the second assigns a high probability of transitioning to distant points, and the third one uses an uniform constant transition matrix as described before. The results are averaged over 10 random seeds.

6 Empirical Studies

This section has two primary objectives: first, to validate whether the theoretical results of variance analysis are applicable to real-world datasets in linear setting; second, to investigate the practical utility of our algorithm on real world dataset. Appendix C.5 shows results on common regression and image classification tasks. Results on additional datasets (e.g. house price, bike rental, weather, insurance) and any missing details are in Appendix C.

Baselines and naming rules. TDReg: our TD approach, with its direct competitor being Reg (conventional l_2 regression). Reg-WP: Utilizes the same probability transition matrix as TDReg but does not employ bootstrap targets. This baseline can be used to assess the effect of bootstrap and transition probability matrix.

Linear Setting with Correlated Noise. As outlined in Section 4.2, we anticipate that in scenarios where the target noise exhibits positive correlation, TD learning should leverage the advantages of using a transition matrix. When this matrix transitions between points with correlated noise, the TD target counterbalances the noise, thereby reducing variance. As depicted in Figure 2, we observe that with an increasing correlation coefficient (indicative of progressively positively correlated noise), TD consistently shows improved generalization performance towards the underlying best baseline. Alongside TD and OLS, we include two baselines known for their efficacy in handling correlated noise: generalized least squares (GLS) and feasible GLS (FGLS) (Aitken, 1936). It is noteworthy that GLS requires fully known covariance of noise generation. FGLS has two procedures: estimate the covariance matrix followed by applying this estimation to resolve the linear system. Please refer to Appendix C.2 for details.

Deep Learning without Synthetic Noise. In order to test the practical utility of our algorithm, we evaluate it using various transition matrices on an air quality dataset (Vito, 2016), without introducing any synthetic noise to alter the original data. This dataset's task is to predict CO concentration. The main objective here is to demonstrate that real-world problems might also feature positively correlated data, where our method could outperform SGD. We provide intuition regarding why the noise on this dataset may be correlated in Appendix C.6. Figure 3 shows the learning curves of TD and baselines using different types of transition matrix P. Our algorithm (TDReg) performs best with a transition matrix P that favors nearby point transitions, but not as well with opposite preferences, showing modest gains over SGD using a uniform matrix. This may be due to the dataset's positive correlation and noise mitigation by the bootstrap target. Comparisons with baseline Reg-WP, using the same matrix to evaluate the bootstrap effect, indicate that TDReg's advantage isn't solely from the matrix, as Reg-WP without bootstrap may diverge, likely from increased correlation by P. Testing confirmed divergence wasn't due to high learning rates, as even with lower rates, baselines didn't improve.

7 Conclusion

This paper introduces a universal framework that transforms traditional SL problems into RL ones, proposing a generalized TD algorithm. We identify specific problem sets suitable for these algorithms and discuss the potential for variance reduction. The paper confirms the convergence properties of our algorithm through empirical tests in linear and deep learning settings, supporting our theoretical claims and assessing practical applicability. This research bridges SL and RL paradigms, offering a new perspective that interconnects various data points.

Future work and limitations. Our covariance analysis does not consider the effects of a non-symmetric matrix $\mathbf{S}$, lacking in both related literature and interpretable expressions. We also have not explored recent TD algorithms like emphatic TD Sutton et al. (2016), or gradient TD Maei (2011); Pan et al. (2017b,a), which could improve stability and convergence. Future research could also assess the utility of the transition matrix in broader applications like transfer learning, domain adaptation, or continual learning, and extend our methods to modern neural networks like transformers to appeal to a wider audience.

References

Abadi, M., Agarwal, A., Barham, P., and et al. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

Aitken, A. C. Iv.—on least squares and linear combination of observations. *Proceedings of the Royal Society of Edinburgh*, pp. 42–48, 1936.

Baltagi, B. H. *Econometrics*. Springer Books. Springer, 2008.

Banerjee, A., Merugu, S., Dhillon, I. S., Ghosh, J., and Lafferty, J. Clustering with bregman divergences. *Journal of machine learning research*, 6(10), 2005.

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

Bishop, C. M. *Pattern Recognition and Machine Learning*. Springer, 2006.

Bradtke, S. J. and Barto, A. G. Linear least-squares algorithms for temporal difference learning. *Machine Learning*, 1996.

Company, T. Travel insurance data, 2021. URL `https://www.kaggle.com/datasets/tejashvi14/travel-insurance-prediction-data/data`.

Fanaee-T, H. and Gama, J. Event labeling combining ensemble detectors and background knowledge. *Progress in Artificial Intelligence*, pp. 1–15, 2013.

Ghosh, D., Gupta, A., Reddy, A., Fu, J., Devin, C. M., Eysenbach, B., and Levine, S. Learning to reach goals via iterated supervised learning. In *International Conference on Learning Representations*, 2021.

Greville, T. N. E. Note on the generalized inverse of a matrix product. *Siam Review*, 8(4):518–521, 1966.

Gupta, D., Mihucz, G., Schlegel, M., Kostas, J., Thomas, P. S., and White, M. Structural credit assignment in neural networks using reinforcement learning. In *Advances in Neural Information Processing Systems*, pp. 30257–30270, 2021.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. *Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016.

Helmbold, D. P., Kivinen, J., and Warmuth, M. K. Relative loss bounds for single neurons. *IEEE Transactions on Neural Networks*, 10(6):1291–1304, 1999.

Hussein, A., Gaber, M. M., Elyan, E., and Jayne, C. Imitation learning: A survey of learning methods. *ACM Comput. Surv.*, 2017.

Joe Young (Owner), A. E. Rain in Australia, Copyright Commonwealth of Australia 2010, Bureau of Meteorology. https://www.kaggle.com/datasets/jsphyg/weather-dataset-rattle-package, 2020. URL `http://www.bom.gov.au/climate/dwo/,http://www.bom.gov.au/climate/data.`

Kariya, T. and Kurata, H. *Generalized least squares*. John Wiley & Sons, 2004.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *International Conference on Learning Representations*, 2015.

Krizhevsky, A. Learning multiple layers of features from tiny images. Technical report, 2009.

Kubát, M. and Matwin, S. Addressing the curse of imbalanced training sets: One-sided selection. *International Conference on Machine Learning*, 1997.

LeCun, Y., Cortes, C., and Burges, C. Mnist handwritten digit database. *ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist*, 2, 2010.

Lee, L., Eysenbach, B., Salakhutdinov, R., Gu, S. S., and Finn, C. Weakly-supervised reinforcement learning for controllable behavior. In *Advances in Neural Information Processing Systems*, 2020.

Lichman, M. UCI machine learning repository, 2015. URL `http://archive.ics.uci.edu/ml`.

Lin, E., Chen, Q., and Qi, X. Deep reinforcement learning for imbalanced classification. *Applied Intelligence*, pp. 2488–2502, 2020.

MacGlashan, J., Ho, M. K., Loftin, R., Peng, B., Wang, G., Roberts, D. L., Taylor, M. E., and Littman, M. L. Interactive learning from policy-dependent human feedback. In *International Conference on Machine Learning*, pp. 2285–2294, 2017.

Maei, H. R. Gradient temporal-difference learning algorithms. *PhD thesis, University of Alberta Education and Research Archive*, 2011.

McCullagh, P. and Nelder, J. A. *Generalized Linear Models*, volume 37. CRC Press, 1989.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., and et al. Human-level control through deep reinforcement learning. *Nature*, 2015.

Nelder, J. A. Log linear models for contingency tables: A generalization of classical least squares. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, pp. 323–329, 1974.

Nelder, J. A. and Wedderburn, R. W. M. Generalized linear models. *Journal of the Royal Statistical Society. Series A (General)*, pp. 370–384, 1972.

Pan, Y., Azer, E. S., and White, M. Effective sketching methods for value function approximation. *Conference on Uncertainty in Artificial Intelligence*, 2017a.

Pan, Y., White, A., and White, M. Accelerated gradient temporal difference learning. *AAAI Conference on Artificial Intelligence*, pp. 2464–2470, 2017b.

Pan, Y., Imani, E., Farahmand, A.-m., and White, M. An implicit function learning approach for parametric modal regression. *Advances in Neural Information Processing Systems*, 33:11442–11452, 2020.

Paredes, E. and Ballester-Ripoll, R. SGEMM GPU kernel performance. UCI Machine Learning Repository, 2018.

Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., and Lerer, A. Automatic differentiation in pytorch. 2017.

Perdomo, J., Zrnic, T., Mendler-Dünner, C., and Hardt, M. Performative prediction. *International Conference on Machine Learning*, pp. 7599–7609, 2020.

Puterman, M. L. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014.

Seabold, S. and Perktold, J. statsmodels: Econometric and statistical modeling with python. In *9th Python in Science Conference*, 2010.

Sutton, R. S. Learning to predict by the methods of temporal differences. *Machine Learning*, pp. 9–44, 1988.

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

Szepesvari, C. *Algorithms for Reinforcement Learning*. Morgan & Claypool Publishers, 2010.

Tsitsiklis, J. N. and Van Roy, B. An analysis of temporal-difference learning with function approximation. *IEEE Transactions on Automatic Control*, 42(5), 1997.

Vito, S. Air Quality. UCI Machine Learning Repository, 2016.

Wang, F., Li, P., and Konig, A. C. Learning a bi-stochastic data similarity matrix. In *2010 IEEE International Conference on Data Mining*, pp. 551–560. IEEE, 2010.

Xiao, H., Rasul, K., and Vollgraf, R. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *CoRR*, 2017. URL `http://arxiv.org/abs/1708.07747`.

Table of Contents

A Proofs

A.1 Proof for Proposition 4.1

Proposition 4.1. *[Connection between TD and OLS] When $\mathbf{D}$ has full support and $\mathbf{X}$ has linearly independent rows, TD and OLS have the same minimum norm solution. Moreover, any solution to the linear system $\mathbf{X}\mathbf{w} = \mathbf{y}$ must be also an solution to TD's linear system $\mathbf{A}\mathbf{w} = \mathbf{b}$ as defined in Equation* (5)*.*

Proof. In our TD formulation, the reward is $\mathbf{r} = (\mathbf{I} - \gamma\mathbf{P})\mathbf{y}$. With $\mathbf{S} = \mathbf{D}(\mathbf{I} - \gamma\mathbf{P})$, we have $\mathbf{A} = \mathbf{X}^\top\mathbf{S}\mathbf{X}$, and $\mathbf{b} = \mathbf{X}^\top\mathbf{S}\mathbf{y}$. To verify the first claim, define The min-norm solutions found by TD and OLS are respectively $\mathbf{w}_{TD} = \mathbf{A}^\dagger\mathbf{b}$, and $\mathbf{w}_{LS} = \mathbf{X}^\dagger\mathbf{y}$, where

$$\mathbf{w}_{TD} = \mathbf{A}^\dagger\mathbf{b} = (\mathbf{X}^\top\mathbf{S}\mathbf{X})^\dagger \cdot \mathbf{X}^\top\mathbf{S}\mathbf{y}. \tag{16}$$

When $\mathbf{D}$ has full support, $\mathbf{S}$ is invertible (thus has linearly independent rows/columns). Additionally, $\mathbf{X}$ has linearly independent rows so $(\mathbf{X}^\top\mathbf{S}\mathbf{X})^\dagger = \mathbf{X}^\dagger\mathbf{S}^{-1}(\mathbf{X}^\top)^\dagger$ (Greville, 1966, Thm.3) and the TD solution becomes

$$\mathbf{w}_{TD} = \mathbf{A}^\dagger\mathbf{b} = \mathbf{X}^\dagger\mathbf{S}^{-1}(\mathbf{X}^\top)^\dagger\mathbf{X}^\top\mathbf{S}\mathbf{y} \tag{17}$$

Finally, when $\mathbf{X}$ has linearly independent rows, $(\mathbf{X}^\top)^\dagger\mathbf{X}^\top = \mathbf{I}_n$ so $\mathbf{w}_{TD} = \mathbf{X}^\dagger\mathbf{y} = \mathbf{w}_{LS}$.

To verify the second part, the TD's linear system is

$$\mathbf{X}^\top\mathbf{S}\mathbf{X}\mathbf{w} = \mathbf{X}^\top\mathbf{S}\mathbf{y}, \tag{18}$$

which is essentially preconditioning the linear system $\mathbf{X}\mathbf{w} = \mathbf{y}$ by $\mathbf{X}^\top\mathbf{S}$. Hence, any solution to the latter is also an solution to TD. □

A.2 Proof for Proposition 4.2

Proof. Recall from Equation (5) that $\mathbf{w}_{TD} = \mathbf{A}^{-1}\mathbf{b}, \mathbf{b} = \mathbf{X}^\top\mathbf{D}\mathbf{r}$, and $\mathbf{r} = (\mathbf{I} - \gamma\mathbf{P})\mathbf{y}$. Therefore $\mathbf{w}_{TD} - \mathbf{w}^*$ equals

$$\begin{aligned}\mathbf{w}_{TD} - \mathbf{w}^* &= \mathbf{A}^{-1}(\mathbf{b} - \mathbf{A}\mathbf{w}^*) = \mathbf{A}^{-1}(\mathbf{X}^\top\mathbf{D}\mathbf{r} - \mathbf{A}\mathbf{w}^*) &(19)\\ &= \mathbf{A}^{-1}(\mathbf{X}^\top\mathbf{D}(\mathbf{I} - \gamma\mathbf{P})\mathbf{y} - \mathbf{A}\mathbf{w}^*) &(20)\\ &= \mathbf{A}^{-1}(\mathbf{X}^\top\mathbf{S}\mathbf{y} - \mathbf{X}^\top\mathbf{S}\mathbf{X}\mathbf{w}^*) = \mathbf{A}^{-1}\mathbf{X}\mathbf{S}\boldsymbol{\epsilon} &(21)\end{aligned}$$

where we define $\mathbf{S} \overset{\text{def}}{=} \mathbf{D}(\mathbf{I} - \gamma\mathbf{P})$. When $\mathbb{E}[\boldsymbol{\epsilon}|\mathbf{X}] = \mathbf{0}$, the conditional expectation of the above equation is zero. Thus $\mathbf{w}_{TD}$ is conditionally unbiased and its conditional covariance is

$$\begin{aligned}\mathrm{Cov}[\mathbf{w}_{TD}|\mathbf{X}] &= \mathbf{A}^{-1}\mathbf{X}^\top\mathbf{S} \cdot \mathrm{Cov}[\boldsymbol{\epsilon}|\mathbf{X}] \cdot \mathbf{S}^\top\mathbf{X}(\mathbf{A}^{-1})^\top &(22)\\ &= \mathbf{A}^{-1}\mathbf{X}^\top\mathbf{S} \cdot \mathbf{C} \cdot \mathbf{S}^\top\mathbf{X}(\mathbf{A}^{-1})^\top &(23)\end{aligned}$$

Finally, when $\mathbf{S} = \mathbf{S}^\top$ and $\mathbf{C} = c \cdot \mathbf{S}^{-1}$ for some $c > 0$, the TD estimator and its covariance become

$$\mathbf{w}_{TD} = (\mathbf{X}^\top\mathbf{S}\mathbf{X})^{-1}\mathbf{X}\mathbf{S}\mathbf{y},\ \mathrm{Cov}[\mathbf{w}_{TD}|\mathbf{X}] = (c\mathbf{X}^\top\mathbf{S}\mathbf{X})^{-1} \tag{24}$$

By using the Cholesky decomposition $\mathbf{S} = \mathbf{L}\mathbf{L}^\top$, one can see that the TD estimator is equivalent to the OLS solution to a rescaled problem $\widetilde{\mathbf{y}} = \widetilde{\mathbf{X}}\mathbf{w} + \widetilde{\boldsymbol{\epsilon}}$ where

$$\widetilde{\mathbf{y}} = \mathbf{L}\mathbf{y} \qquad \widetilde{\mathbf{X}} = \mathbf{L}\mathbf{X} \qquad \widetilde{\boldsymbol{\epsilon}} = \mathbf{L}\boldsymbol{\epsilon} \tag{25}$$

Here $\mathrm{Cov}[\widetilde{\boldsymbol{\epsilon}}|\mathbf{X}] = \mathbf{L}\mathbf{C}\mathbf{L}^\top = c\mathbf{I}$ so the TD estimator is the OLS solution to this problem and thus is the BLUE. □

A.3 Proof for Proposition 4.3

Proposition 4.3. *[Variance] Assume the estimated next-state value $\widehat{y}_{t+1}$ satisfies $\widehat{y}_{t+1} = \mathbb{E}[y_{t+1}] + \epsilon$ where ϵ is some independent noise with zero mean and standard deviation σ_ϵ. Let σ_t be the standard deviation of y_t and $\rho_{t,t+1}$ be the Pearson correlation coefficient between y_t and y_{t+1}. If $\rho_{t,t+1} \geq \frac{\gamma^2\sigma_{t+1}^2 + \sigma_\epsilon^2}{2\gamma\sigma_t\sigma_{t+1}}$, then $\mathrm{Var}(y_{t,td}) \leq \mathrm{Var}(y_t)$.*

Proof. We rewrite the TD target (9) as

$$y_{t,td} = (y_t - \gamma y_{t+1}) + \gamma\mathbb{E}[y_{t+1}] + \epsilon = y_t - \gamma(y_{t+1} - \mathbb{E}[y_{t+1}]) + \epsilon$$

This means we can treat y_{t+1} as a control variate and the variance of this estimate is

$$\begin{aligned}\mathrm{Var}(y_{t,td}) &= \mathrm{Var}(y_t) + \gamma^2\mathrm{Var}(y_{t+1}) - 2\gamma\mathrm{Cov}(y_t, y_{t+1}) + \mathrm{Var}(\epsilon)\\ &= \sigma_t^2 + \gamma^2\sigma_{t+1}^2 - 2\gamma\rho_{t,t+1}\sigma_t\sigma_{t+1} + \sigma_\epsilon^2\end{aligned}$$

Plugging into the condition on $\rho_{t,t+1}$ would yield $\mathrm{Var}(y_{t,td}) \leq \mathrm{Var}(y_t)$. □

A.4 Convergence under Expected Update

We first provide assumptions and lemma below.

Assumption A.1 (Feature regularity)**.** $\forall s \in \mathcal{S}$, $\|\mathbf{x}(s)\|_2 \leq 1$ and the steady-state covariance matrix $\mathbf{\Sigma} \overset{\text{def}}{=} D(s)\mathbf{x}(s)\mathbf{x}(s)^\top$ has full rank.

Assumption A.1 is a typical assumption necessary for the existence of the fixed point when there is no transform function (Tsitsiklis & Van Roy, 1997).

Assumption A.2. The inverse link function $f : \mathbb{R} \mapsto \mathcal{Y}$ is continuous, invertible and strictly increasing. Moreover, it has bounded derivative f' on any bounded domain.

Remark. Assumption A.2 is satisfied for those inverse link functions commonly used in GLMs, including but not limit to identity function (linear regression), exponential function (Poisson regression), and sigmoid function (logistic regression) (McCullagh & Nelder, 1989, Table 2.1). We can consider training in a sufficiently large compact $\mathcal{W} \subset \mathbb{R}^d$ such that the fixed point $\mathbf{w}^* \in \mathcal{W}$. Then the following lemma holds.

Lemma A.3. *Under Assumption A.1, A.2 and $\mathbf{w} \in \mathcal{W}$, there exists $L \geq 1$ such that $\forall s_1, s_2 \in \mathcal{S}$ with $z_1 = \mathbf{x}(s_1)^\top\mathbf{w}, z_2 = \mathbf{x}(s_2)^\top\mathbf{w}$,*

$$\frac{1}{L}|z_1 - z_2| \leq |f(z_1) - f(z_2)| \leq L|z_1 - z_2|. \tag{26}$$

The next assumption is necessary later to ensure that the step size is positive.

Assumption A.4 (Bounded discount)**.** The discount factor satisfies $\gamma < \frac{1}{L^2}$ for the L in Lemma A.3.

The convergence proofs resemble those in Bhandari et al. (2018), adapted to handle our specific case with a transformation function f.

Lemma A.3. *Under Assumption A.1, A.2 and $\mathbf{w} \in \mathcal{W}$, there exists $L \geq 1$ such that $\forall s_1, s_2 \in \mathcal{S}$ with $z_1 = \mathbf{x}(s_1)^\top\mathbf{w}, z_2 = \mathbf{x}(s_2)^\top\mathbf{w}$,*

$$\frac{1}{L}|z_1 - z_2| \leq |f(z_1) - f(z_2)| \leq L|z_1 - z_2|. \tag{26}$$

Proof. Assumption A.1 and the compactness of $\mathcal{W}$ ensure that the linear prediction $z = \mathbf{x}(s)^\top\mathbf{w}, \forall s \in \mathcal{S}$ will also be in a compact domain. Given that f is continuous and invertible

(Assumption A.2), effectively the domain and image of will also be compact. Furthermore, since f' is bounded, there exists a constant $L = \max(L_f, L_{f^{-1}}) \geq 1$ such that Equation (26) holds $\forall s_1, s_2 \in \mathcal{S}$ with $z_1 = \mathbf{x}(s_1)^\top \mathbf{w}, z_2 = \mathbf{x}(s_2)^\top \mathbf{w}$, where $L_f, L_{f^{-1}}$ are the Lipschitz constants of f and f^{-1} respectively. □

As mentioned in the main text, the strategy is to bound the second and third terms of the RHS of

$$\|\mathbf{w}^* - \mathbf{w}_{t+1}\|_2^2 = \|\mathbf{w}^* - \mathbf{w}_t\|_2^2 - 2\alpha(\mathbf{w}^* - \mathbf{w}_t)^\top \overline{g}(\mathbf{w}_t) + \alpha^2 \|\overline{g}(\mathbf{w}_t)\|_2^2 \tag{27}$$

Denote $y^* = f(z^*) = f(\mathbf{x}^\top \mathbf{w}^*)$ and $y^*_{td} = f(z^*_{td}) = f(r + \mathbf{x}'^\top \mathbf{w}^*)$. The next two lemmas bound the second and third terms respectively.

Lemma A.5. *For* $\mathbf{w} \in \mathcal{W}$, $(\mathbf{w}^* - \mathbf{w})^\top \overline{g}(\mathbf{w}) \geq \left(\frac{1}{L} - \gamma L\right) \cdot \mathbb{E}\left[(z^* - z)^2\right]$

Proof. Note that

$$\begin{aligned}(\mathbf{w}^* - \mathbf{w})^\top \overline{g}(\mathbf{w}) &= (\mathbf{w}^* - \mathbf{w})^\top [\overline{g}(\mathbf{w}) - \overline{g}(\mathbf{w}^*)] && (28)\\ &= (\mathbf{w}^* - \mathbf{w})^\top \mathbb{E}[\,[(y_{td} - y) - (y^*_{td} - y^*)]\mathbf{x}\,] && (29)\\ &= \mathbb{E}[\,(z^* - z) \cdot [(y^* - y) - (y^*_{td} - y_{td})]\,] && (30)\end{aligned}$$

By Lemma A.3 and using the assumption that f is strictly increasing, we have $(z^* - z)(y^* - y) \geq \frac{1}{L}(z^* - z)^2$. Moreover, the function $\mathbf{z} \mapsto f(\mathbf{r} + \gamma \mathbf{P}\mathbf{z})$ is (γL)-Lipschitz so

$$\mathbb{E}\left[\,(z^* - z)(y^*_{td} - y_{td})\,\right] \leq \gamma L \cdot \mathbb{E}[(z^* - z)^2]. \tag{31}$$

Plug these two to Equation (30) completes the proof. □

Lemma A.6. *For* $\mathbf{w} \in \mathcal{W}$, $\|\overline{g}(\mathbf{w})\|_2 \leq 2L\sqrt{\mathbb{E}\left[(z^* - z)^2\right]}$

Proof. To start

$$\begin{aligned}\|\overline{g}(\mathbf{w})\|_2 &= \|\overline{g}(\mathbf{w}) - \overline{g}(\mathbf{w}^*)\|_2 && && (32)\\ &= \|\mathbb{E}[\,[(y_{td} - y) - (y^*_{td} - y^*)]\mathbf{x}\,]\|_2 && && (33)\\ &\leq \sqrt{\mathbb{E}[\|\mathbf{x}\|_2^2]}\sqrt{\mathbb{E}\left[[(y_{td} - y) - (y^*_{td} - y^*)]^2\right]} && \text{Cauchy-Schwartz} && (34)\\ &\leq \sqrt{\mathbb{E}\left[[(y^* - y) - (y^*_{td} - y_{td})]^2\right]} && \text{Assumption A.1} && (35)\end{aligned}$$

Let $\delta \overset{\text{def}}{=} y^* - y$ and $\delta_{td} \overset{\text{def}}{=} y^*_{td} - y_{td}$, then

$$\begin{aligned}\mathbb{E}\left[(\delta - \delta_{td})^2\right] &= \mathbb{E}[\delta^2] + \mathbb{E}[\delta_{td}^2] - 2\mathbb{E}[\delta\delta_{td}] && && (36)\\ &\leq \mathbb{E}[\delta^2] + \mathbb{E}[\delta_{td}^2] + 2|\mathbb{E}[\delta\delta_{td}]| && && (37)\\ &\leq \mathbb{E}[\delta^2] + \mathbb{E}[\delta_{td}^2] + 2\sqrt{\mathbb{E}[\delta^2]\mathbb{E}[\delta_{td}^2]} && \text{Cauchy-Schwartz} && (38)\\ &= \left(\sqrt{\mathbb{E}[\delta^2]} + \sqrt{\mathbb{E}[\delta_{td}^2]}\right)^2 && && (39)\end{aligned}$$

As a result,

$$\begin{aligned}\|\overline{g}(\mathbf{w})\|_2 &\leq \sqrt{\mathbb{E}\left[(y^* - y)^2\right]} + \sqrt{\mathbb{E}\left[(y^*_{td} - y_{td})^2\right]} && (40)\\ &\leq L\left\{\sqrt{\mathbb{E}\left[(z^* - z)^2\right]} + \sqrt{\mathbb{E}\left[(z^*_{td} - z_{td})^2\right]}\right\} && (41)\end{aligned}$$

Finally, note that

$$\begin{aligned}\mathbb{E}\left[(z^*_{td} - z_{td})^2\right] &= \mathbb{E}\left[\,[(r + \gamma\mathbf{x}'^\top \mathbf{w}^*) - (r + \gamma\mathbf{x}'^\top \mathbf{w})]^2\,\right] && (42)\\ &= \gamma^2 \mathbb{E}\left[\,(\mathbf{x}'^\top \mathbf{w}^* - \mathbf{x}'^\top \mathbf{w})^2\,\right] && (43)\\ &= \gamma^2 \mathbb{E}\left[\,(z^* - z)^2\,\right] && (44)\end{aligned}$$

where the last line is because both s, s' are assumed to be from the stationary distribution. Plugging this to Equation (41) and use the fact that $\gamma \leq 1$ complete the proof. □

Now we are ready to prove the main theorem:

Theorem 5.1. *[Convergence with Expected Update] Under Assumption A.1-A.4, consider the sequence* $(\mathbf{w}_0, \mathbf{w}_1, \cdots)$ *satisfying Equation* (11)*. Let* $\overline{\mathbf{w}}_T \overset{def}{=} \frac{1}{T}\sum_{t=0}^{T-1} \mathbf{w}_t$*,* $\overline{z}_T = \mathbf{x}^\top \overline{\mathbf{w}}_T$ *and* $z^* = \mathbf{x}^\top \mathbf{w}^*$*. By choosing* $\alpha = \frac{1-\gamma L^2}{4L^3} > 0$*, we have*

$$\mathbb{E}\left[(z^* - \overline{z}_T)^2\right] \leq \left(\frac{2L^2}{1-\gamma L^2}\right)^2 \frac{\|\mathbf{w}^* - \mathbf{w}_0\|_2^2}{T} \tag{12}$$

$$\|\mathbf{w}^* - \mathbf{w}_T\|_2^2 \leq \exp\left(-T\omega\left(\frac{1-\gamma L^2}{2L^2}\right)^2\right)\|\mathbf{w}^* - \mathbf{w}_0\|_2^2 \tag{13}$$

Proof. With probability 1, for any $t \in \mathbb{N}_0$

$$\begin{aligned}\|\mathbf{w}^* - \mathbf{w}_{t+1}\|_2^2 &= \|\mathbf{w}^* - \mathbf{w}_t\|_2^2 - 2\alpha(\mathbf{w}^* - \mathbf{w}_t)^\top \overline{g}(\mathbf{w}_t) + \alpha^2\|\overline{g}(\mathbf{w}_t)\|_2^2 &(45)\\ &\leq \|\mathbf{w}^* - \mathbf{w}_t\|_2^2 - \left(2\alpha\left(\frac{1}{L} - \gamma L\right) - 4L^2\alpha^2\right)\mathbb{E}\left[(z^* - z_t)^2\right] &(46)\end{aligned}$$

where $z_t \overset{\text{def}}{=} \mathbf{x}^\top \mathbf{w}_t$. Using $\alpha = \frac{1-\gamma L^2}{4L^3} > 0$

$$\|\mathbf{w}^* - \mathbf{w}_{t+1}\|_2^2 \leq \|\mathbf{w}^* - \mathbf{w}_t\|_2^2 - \left(\frac{1-\gamma L^2}{2L^2}\right)^2 \mathbb{E}\left[(z^* - z_t)^2\right] \tag{47}$$

Telescoping sum gives

$$\left(\frac{1-\gamma L^2}{2L^2}\right)^2 \times \sum_{t=0}^{T-1}\mathbb{E}\left[(z^* - z_t)^2\right] \leq \sum_{t=0}^{T-1}(\|\mathbf{w}^* - \mathbf{w}_t\|_2^2 - \|\mathbf{w}^* - \mathbf{w}_{t+1}\|_2^2) \leq \|\mathbf{w}^* - \mathbf{w}_0\|_2^2 \tag{48}$$

By Jensen's inequality

$$\mathbb{E}\left[(z^* - \overline{z}_T)^2\right] \leq \frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\left[(z^* - z_t)^2\right] \leq \left(\frac{2L^2}{1-\gamma L^2}\right)^2 \frac{\|\mathbf{w}^* - \mathbf{w}_0\|_2^2}{T} \tag{49}$$

Finally since we assume that $\|\mathbf{x}(s)\|_2^2 \leq 1, \forall s$, we have $\mathbb{E}\left[(z^* - z_t)^2\right] \geq \omega\|\mathbf{w}^* - \mathbf{w}_t\|_2^2$ where ω is the maximum eigenvalue of the steady-state feature covariance matrix $\mathbf{\Sigma} = \mathbf{X}^\top\mathbf{D}\mathbf{X} = \sum_s D(s)\mathbf{x}(s)\mathbf{x}(s)^\top$. Therefore, Equation (47) leads to

$$\begin{aligned}\|\mathbf{w}^* - \mathbf{w}_{t+1}\|_2^2 &\leq \left(1 - \omega\left(\frac{1-\gamma L^2}{2L^2}\right)^2\right)\|\mathbf{w}^* - \mathbf{w}_t\|_2^2 &(50)\\ &\leq \exp\left(-\omega\left(\frac{1-\gamma L^2}{2L^2}\right)^2\right)\|\mathbf{w}^* - \mathbf{w}_t\|_2^2 \qquad \forall x \in \mathbb{R}, 1 - x \leq e^{-x} &(51)\end{aligned}$$

Repeatedly applying this bound gives Equation (13). □

A.5 Convergence under Sample-based Update

To account for the randomness, let $\sigma^2 \overset{\text{def}}{=} \mathbb{E}[\|g_t(\mathbf{w}^*)\|_2^2]$, the variance of the TD update at the stationary point $\mathbf{w}^*$ under the stationary distribution. Similar to Lemma A.6, the following lemma bounds the expected norm of the update:

Lemma A.7. *For* $\mathbf{w} \in \mathcal{W}$*,* $\mathbb{E}[\|g_t(\mathbf{w})\|_2^2] \leq 2\sigma^2 + 8L^2\mathbb{E}[(z_t^* - z_t)^2]$ *where* $\sigma^2 = \mathbb{E}[\|g_t(\mathbf{w}^*)\|_2^2]$*.*

Proof. To start

$$
\begin{aligned}
\mathbb{E}[\|g_t(\mathbf{w})\|_2^2] &= \mathbb{E}[\|g_t(\mathbf{w}) - g_t(\mathbf{w}^*) + g_t(\mathbf{w}^*)\|_2^2] && \tag{52}\\
&\leq \mathbb{E}[(\|g_t(\mathbf{w}^*)\|_2 + \|g_t(\mathbf{w}) - g_t(\mathbf{w}^*)\|_2)^2] && \text{Triangle inequality} \tag{53}\\
&\leq 2\mathbb{E}[(\|g_t(\mathbf{w}^*)\|_2^2] + 2\mathbb{E}[\|g_t(\mathbf{w}) - g_t(\mathbf{w}^*)\|_2^2] && (a+b)^2 \leq 2a^2 + 2b^2 \tag{54}\\
&\leq 2\sigma^2 + 2\mathbb{E}\left[\ \|[(y_{t,td} - y_t) - (y_{t,td}^* - y_t^*)]\mathbf{x}_t\|_2^2\ \right] && \tag{55}\\
&\leq 2\sigma^2 + 2\mathbb{E}\left[\ ((y_{t,td} - y_t) - (y_{t,td}^* - y_t^*))^2\ \right] && \text{Assumption A.1} \tag{56}\\
&= 2\sigma^2 + 2\mathbb{E}\left[\ ((y_t^* - y_t) - (y_{t,td}^* - y_{t,td}))^2\ \right] && \tag{57}\\
&\leq 2\sigma^2 + 4\left(\mathbb{E}[(y_t^* - y_t)^2] + \mathbb{E}[(y_{t,td}^* - y_{t,td})^2]\right) && (a-b)^2 \leq 2a^2 + 2b^2 \tag{58}
\end{aligned}
$$

where $y_t^* \overset{\text{def}}{=} f(z_t^*) = f(\mathbf{x}_t^\top \mathbf{w}^*)$ and $y_{t,td}^* \overset{\text{def}}{=} f(z_{t,td}^*) = f(r_t + \mathbf{x}_{t+1}^\top \mathbf{w}^*)$. Note that by Lemma A.3

$$
\mathbb{E}[(y_t^* - y_t)^2] + \mathbb{E}[(y_{t,td}^* - y_{t,td})^2] \leq L^2\left\{\mathbb{E}\left[(z_t^* - z_t)^2\right] + \mathbb{E}\left[(z_{t,td}^* - z_{t,td})^2\right]\right\}. \tag{59}
$$

Finally,

$$
\begin{aligned}
\mathbb{E}\left[(z_{t,td}^* - z_{t,td})^2\right] &= \mathbb{E}\left[\ [(r_t + \gamma\mathbf{x}_{t+1}^\top \mathbf{w}^*) - (r_t + \gamma\mathbf{x}_{t+1}^\top \mathbf{w}_t)]^2\ \right] \tag{60}\\
&= \gamma^2\mathbb{E}\left[\ (\mathbf{x}_{t+1}^\top \mathbf{w}^* - \mathbf{x}_{t+1}^\top \mathbf{w}_t)^2\ \right] \tag{61}\\
&= \gamma^2\mathbb{E}\left[\ (z_t^* - z_t)^2\ \right] \tag{62}
\end{aligned}
$$

where the last line is because both s_t, s_{t+1} are from the stationary distribution. Combining these with Equation (58) gives

$$
\begin{aligned}
\mathbb{E}[\|g_t(\mathbf{w})\|_2^2] &\leq 2\sigma^2 + 4L^2(1+\gamma^2)\mathbb{E}\left[(z_t^* - z_t)^2\right] && \tag{63}\\
&\leq 2\sigma^2 + 8L^2\mathbb{E}\left[(z_t^* - z_t)^2\right]. && 0 \leq \gamma \leq 1 \tag{64}
\end{aligned}
$$

□

Now we are ready to present the convergence when using i.i.d. sample for the update:

Theorem 5.2. *[Convergence with Sampled-based Update] Under Assumption A.1-A.4, with sample-based update Equation* (14)*, let* $\sigma^2 = \mathbb{E}[\|g_t(\mathbf{w}^*)\|_2^2]$*,* $\overline{\mathbf{w}}_T \overset{def}{=} \frac{1}{T}\sum_{t=0}^{T-1}\mathbf{w}_t$*,* $\overline{z}_T = \mathbf{x}^\top\overline{\mathbf{w}}_T$ *and* $z^* = \mathbf{x}^\top\mathbf{w}^*$*. For* $T \geq \frac{64L^6}{(1-\gamma L^2)^2}$ *and a constant step size* $\alpha_t = 1/\sqrt{T}, \forall t$*, we have*

$$
\mathbb{E}\left[(z^* - \overline{z}_T)^2\right] \leq \frac{L\left(\|\mathbf{w}^* - \mathbf{w}_0\|_2^2 + 2\sigma^2\right)}{\sqrt{T}(1-\gamma L^2)}. \tag{15}
$$

Proof. Note that Lemma A.5 holds for any $\mathbf{w} \in \mathcal{W}$ and the expectation in $\overline{g}(\mathbf{w}) = \mathbb{E}[g_t(\mathbf{w})]$ is based on the sample (s_t, r_t, s_{t+1}), *regardless of the choice of* $\mathbf{w}$. Thus, one can choose $\mathbf{w} = \mathbf{w}_t$ and then $\mathbb{E}[g_t(\mathbf{w}_t)|\mathbf{w}_t] = \overline{g}(\mathbf{w}_t)$. As a result, both Lemma A.5 and Lemma A.7 can be applied to $\mathbb{E}[(\mathbf{w}^* - \mathbf{w}_t)^\top g_t(\mathbf{w}_t)|\mathbf{w}_t]$ and $\mathbb{E}[\|g_t(\mathbf{w}_t)\|_2^2|\mathbf{w}_t]$, respectively, in the following. For any $t \in \mathbb{N}_0$

$$
\begin{aligned}
\mathbb{E}[\|\mathbf{w}^* - \mathbf{w}_{t+1}\|_2^2] &= \mathbb{E}[\|\mathbf{w}^* - \mathbf{w}_t\|_2^2] - 2\alpha_t\mathbb{E}[(\mathbf{w}^* - \mathbf{w}_t)^\top g_t(\mathbf{w}_t)] + \alpha_t^2\mathbb{E}[\|g_t(\mathbf{w}_t)\|_2^2] \tag{65}\\
&= \mathbb{E}[\|\mathbf{w}^* - \mathbf{w}_t\|_2^2] - 2\alpha_t\mathbb{E}[\mathbb{E}[(\mathbf{w}^* - \mathbf{w}_t)^\top g_t(\mathbf{w}_t)|\mathbf{w}_t]] + \alpha_t^2\mathbb{E}[\mathbb{E}[\|g_t(\mathbf{w}_t)\|_2^2|\mathbf{w}_t]] \tag{66}\\
&\leq \mathbb{E}[\|\mathbf{w}^* - \mathbf{w}_t\|_2^2] - \left(2\alpha_t\left(\frac{1}{L} - \gamma L\right) - 8L^2\alpha_t^2\right)\mathbb{E}\left[(z^* - z_t)^2\right] + 2\alpha_t^2\sigma^2 \tag{67}\\
&\leq \mathbb{E}[\|\mathbf{w}^* - \mathbf{w}_t\|_2^2] - \alpha_t\left(\frac{1}{L} - \gamma L\right)\mathbb{E}\left[(z^* - z_t)^2\right] + 2\alpha_t^2\sigma^2 \tag{68}
\end{aligned}
$$

where the last inequality is due to $\alpha_t = \frac{1}{\sqrt{T}} \leq \frac{1-\gamma L^2}{8L^3}$. Then telescoping sum gives

$$
\begin{aligned}
&\frac{1}{\sqrt{T}}\left(\frac{1}{L} - \gamma L\right)\sum_{t=0}^{T-1}\mathbb{E}\left[(z^* - z_t)^2\right] \leq \|\mathbf{w}^* - \mathbf{w}_0\|_2^2 + 2\sigma^2 \tag{69}\\
\iff &\sum_{t=0}^{T-1}\mathbb{E}\left[(z^* - z_t)^2\right] \leq \frac{\sqrt{T}L}{1-\gamma L^2}\left(\|\mathbf{w}^* - \mathbf{w}_0\|_2^2 + 2\sigma^2\right). \tag{70}
\end{aligned}
$$

Finally, Jensen's inequality completes the proof

$$\mathbb{E}\left[(z^* - \bar{z}_T)^2\right] \leq \frac{1}{T}\sum_{t=0}^{T-1} \mathbb{E}\left[(z^* - z_t)^2\right] \leq \frac{L\left(\|\mathbf{w}^* - \mathbf{w}_0\|_2^2 + 2\sigma^2\right)}{\sqrt{T}(1-\gamma L^2)}. \tag{71}$$

□

B Experiment Details: Synthetic Data

B.1 Verifying Equivalence

This subsection provides details of the empirical verification in Section 4.1.

For `Random`, each element of $\mathbf{P}$ is drawn from the uniform distribution $U(0,1)$ and then normalized so that each row sums to one. The `Deficient` variant is exact the same as `Random`, except that the last column is set to all zeros before row normalization. This ensures that the last state is never visited from any state, thus not having full support in its stationary distribution. `Uniform` simply means every element of $\mathbf{P}$ is set to $1/n$ where $n = 100$ is the number of training points. `Distance (Close)` assigns higher transition probability to points closer to the current point in the label space. Finally, `Distance (Far)` means contrary to `Distance (Close)`. The last two variants are used to see if similarity between points in the label space can play a role in the transition when using our TD algorithm.

Each element of the input matrix $\mathbf{X}$ is drawn from the standard normal distribution $\mathcal{N}(0,1)$. This (almost surely) guarantees that $\mathbf{X}$ has linearly independent rows in the overparametrization regime (i.e., when $n < d$). The true model $\mathbf{w}^*$ is set to be a vector of all ones. Each label y_i is generated by $y_i = \mathbf{x}_i^\top \mathbf{w}^* + \epsilon_i$ with noise $\epsilon_i \sim \mathcal{N}(0, 0.1^2)$.

We test various transition matrices $\mathbf{P}$ as shown in Table 2. For `Random`, each element of $\mathbf{P}$ is drawn from the uniform distribution $U(0,1)$ and then normalized so that each row sums to one. The `Deficient` variant is exact the same as `Random`, except that the last column is set to all zeros before row normalization. This ensures that the last state is never visited from any state, thus not having full support in its stationary distribution. `Uniform` simply means every element of $\mathbf{P}$ is set to $1/n$ where $n = 100$ is the number of training points. `Distance (Close)` assigns higher transition probability to points closer to the current point, where the element in the ith row and the jth column is first set to $\exp(-(y_i - y_j)^2/2)$ then the whole matrix is row-normalized. Finally, `Distance (Far)` uses $1 - \exp(-(y_i - y_j)^2/2)$ before normalization. The last two variants are used to see if similarity between points can play a role in the transition when using our TD algorithm.

As shown in Table 2, the min-norm solution $\mathbf{w}_{TD}$ is very close to the min-norm solution of OLS as long as $n < d$ and $\mathbf{D}$ has full support (non-deficient $\mathbf{P}$). The choice of $\mathbf{P}$ only has little effect in such cases. This synthetic experiment verifies our analysis in the main text.

B.2 Positively Correlated Data

This subsection provides details of the empirical verification (Figure 1) in Section 4.2.

In each run, we jointly sample $n = 200$ data points (100 for training and 100 for test) from a Gaussian process where each element of the input matrix $\mathbf{X}$ is drawn from the standard normal distribution $\mathcal{N}(0,1)$. The input dimension is $d = 70$. For the outputs, the mean function is given by $m(\mathbf{x}) = \mathbf{x}^\top \mathbf{w}^*$ where $\mathbf{w}^*$ is a vector of all ones and the covariance matrix is a block diagonal matrix

$$\mathbf{C} = \begin{bmatrix} \widetilde{\mathbf{C}} & & \\ & \ddots & \\ & & \widetilde{\mathbf{C}} \end{bmatrix}_{200\times 200} \quad \text{with} \quad \widetilde{\mathbf{C}} = \begin{bmatrix} 1 & \rho & \dots & \rho \\ \rho & 1 & \dots & \rho \\ \vdots & \vdots & \ddots & \vdots \\ \rho & \rho & \dots & 1 \end{bmatrix}_{10\times 10} \tag{72}$$

where ρ is a tuning parameter for the correlation. When $\rho > 0$, this structure ensures that all ten points within the same "cluster" are positively correlated. Finally, we add an independent noise (with zero mean and standard deviation of 0.1) to each of the output before using them for training and testing.

For our TD method, we set $\gamma = 0.99$ and design the probability matrix as an interpolation between an covariance matrix $\mathbf{C}$ and $\mathbf{1} - \mathbf{C}$, defined as $\mathbf{P} = (1 - \eta)(\mathbf{1} - \mathbf{C}) + \eta\mathbf{C}$, followed by normalization to ensure it forms a valid stochastic matrix. We vary η over the set $\{0.5, 0.6, 0.7, 0.8, 0.9\}$ and the correlation coefficient ρ over $\{0.1, 0.3, 0.5, 0.7, 0.9\}$. With 100 training points, we learn both the TD solution $\mathbf{w}_{TD}$ and OLS solution $\mathbf{w}_{LS}$ and plot their test RMSE. The experiment is repeated 50 times and Figure 1 reports the mean and standard deviation for different (ρ, η) values.

C Experiment Details: Real-world Data

C.1 Implementation Details

Deep learning experiments are based on tensorflow (Abadi et al., 2015), version 2.11.0, except that the ResNN18 experiments are using pytorch (Paszke et al., 2017). Core part of code is available at `https://github.com/yannickycpan/reproduceSL.git`. Below introduce common setup; different settings will be specified when mentioned.

Datasets. We use three popular datasets house price (Lichman, 2015), execution time (Paredes & Ballester-Ripoll, 2018) and Bikeshare (Fanaee-T & Gama, 2013) as benchmark datasets. We have performed one-hot encoding for all categorical variables and removed irrelevant features such as date and year as done by Pan et al. (2020). This preprocessing results in 114 features. The Bikeshare dataset, which uses count numbers as its target variable, is popularly used for testing Poisson regressions. The air quality dataset (Vito, 2016) is loaded by using package *ucimlrepo* by `from ucimlrepo import fetch_ucirepo`. For image classification, we use the popular MNIST (LeCun et al., 2010), Fashion-MNIST (Xiao et al., 2017), Cifar10 and Cifar100 (Krizhevsky, 2009) datasets. In TD algorithms, unless otherwise specified, we use a fixed transition probability matrix with all entries set to $1/n$, which is memory and computation efficient, and simplifies sampling processes.

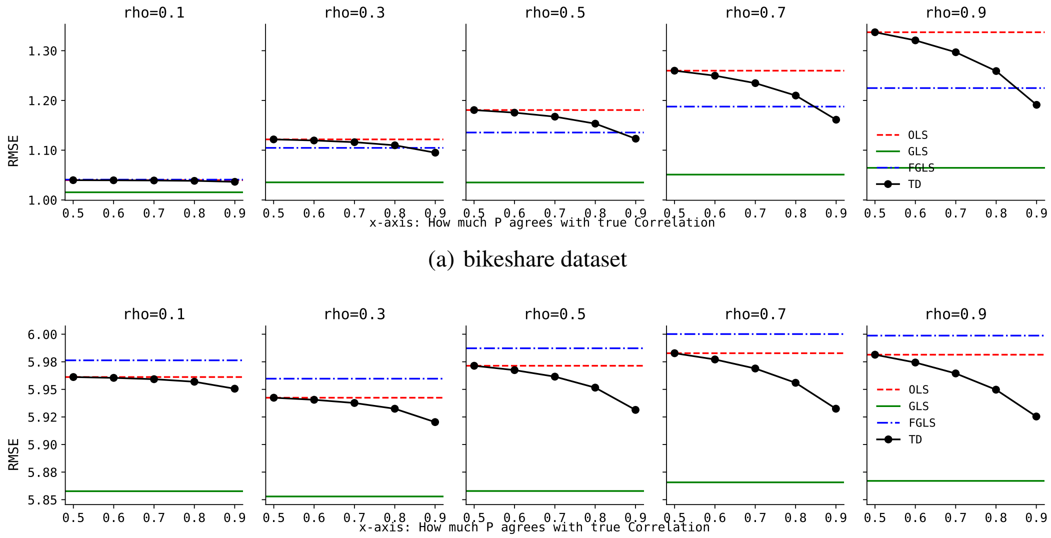
For image datasets, we employ CNN consisting of three convolution layers with the number of kernels 32, 64, and 64, each with a filter size of 2×2. This was followed by two fully connected hidden layers with 256 and 128 units, respectively, before the final output layer. The ResNN18 is imported from pytorch. On all image datasets, Adam optimizer is used and the learning rate sweeps over $\{0.003, 0.001, 0.0003\}$, $\gamma \in \{0.01, 0.1, 0.2\}$, $\tau \in \{0.01, 0.1\}$. The neural network is trained with mini-batch size 128. For ResNN18, we use learning rate 0.001 and $\tau = 0.01$.

Hyperparameter settings. For regression and binary classification tasks, we employ neural networks with two hidden layers of size 256x256 and ReLU activation functions. These networks are trained with a mini-batch size of 128 using the Adam optimizer (Kingma & Ba, 2015). In our TD algorithm, we perform hyperparameter sweeps for $\gamma \in \{0.1, 0.9\}$, target network moving rate $\tau \in \{0.01, 0.1\}$. For all algorithms we sweep learning rate $\alpha \in \{0.0003, 0.001, 0.003, 0.01\}$, except for cases where divergence occurs, such as in Poisson regression on the Bikeshare dataset, where we additionally sweep $\{0.00003, 0.0001\}$. Training iterations are set to 15k for the house data, 25k for Bikeshare, and 30k for other datasets. We perform random splits of each dataset into 60% for training and 40% for testing for each random seed or independent run. Hyperparameters are optimized over the final 25% of evaluations to avoid divergent settings. The reported results in the tables represent the average of the final two evaluations after smoothing window.

Naming rules. For convenience, we repeat naming rules from the main body here. TDReg: our TD approach, with its direct competitor being Reg (conventional l_2 regression). Reg-WP: Utilizes the same probability transition matrix as TDReg but does not employ bootstrap targets. This baseline can be used to assess the effect of bootstrap and transition probability matrix. On Bikesharedata, TDReg uses an exponential link function designed for handling counting data, and the baseline becomes Poisson regression correspondingly.

C.2 Additional Results on Linear Regression

As complementary results to the execute time dataset presented in Section 6, we provide results on two other regression datasets below (see Figure 4). We consistently observe that the TD algorithm performs more closely to the underlying best estimator. Notably, the performance gain tends to increase as the correlation strengthens or as the transition probability matrix better aligns with the data correlation.



(a) bikeshare dataset

(b) house price dataset

Figure 4: Test Root Mean Squared Error (RMSE) versus different values of P on bikeshare and execution time dataset. From left to right, the noise correlation coefficient increases ($\rho \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$). Each plot's x-axis represents the degree of alignment between the transition matrix P and the true covariance matrix that generates the noise. Consistent with our expectations, as P approaches the true correlation – implying a higher likelihood of data points with positively correlated noise transitioning from one to another – the solution derived from TD increasingly approximates the optimal one. Furthermore, as the correlation among the data intensifies, TD's solution is closer to optimum and one can see larger gap between TD and OLS/FGLS. The results are averaged over 30 runs.

When implementing FGLS, we initially run OLS to obtain the residuals. Subsequently, an algorithm is employed to fit these residuals for estimating the noise covariance matrix. This matrix is then utilized to compute the closed-form solution. The implementation is done by API from Seabold & Perktold (2010).

In this set of experiments, to speed up multiple matrix inversion and noise sampling, we randomly take 500 subset of the original datasets for training and testing.

C.3 Additional Results: Regression with Correlated Noise with NNs

Similar to the previous experiment, we introduced noise to the original training target using a GP and opted for a uniform transition matrix. The results, depicted in Figure 5, reveal two key observations: 1) As the noise level increases, the performance advantage of TD over the baselines becomes more pronounced; 2) The baseline Reg-WP performs just as poorly as conventional Reg, underscoring the pivotal role of the TD target in achieving performance gains. Additionally, it was observed that all algorithms select the same optimal learning rate even when smaller learning rates are available, and TD consistently chooses a large $\gamma = 0.95$. This implies that TD's stable performance is attributed NOT to the choice of a smaller learning rate, but rather to the bootstrap target. Similar results on house data is shown in Figure 6.

C.4 Additional Results: Binary Classification with NNs

For binary classification, we utilize datasets from Australian weather (Joe Young , Owner), and travel insurance (Company, 2021). The aim of this series of experiments is to examine: 1) the impact of utilizing a link function; 2) with a specially designed transition probability matrix, the potential unique benefits of the TD algorithm in an imbalanced classification setting. Our findings indicate that: 1) the link function significantly influences performance; and 2) while the transition matrix proves beneficial in addressing class imbalance, this advantage seems to stem primarily from up/down-sampling rather than from the TD bootstrap targets.

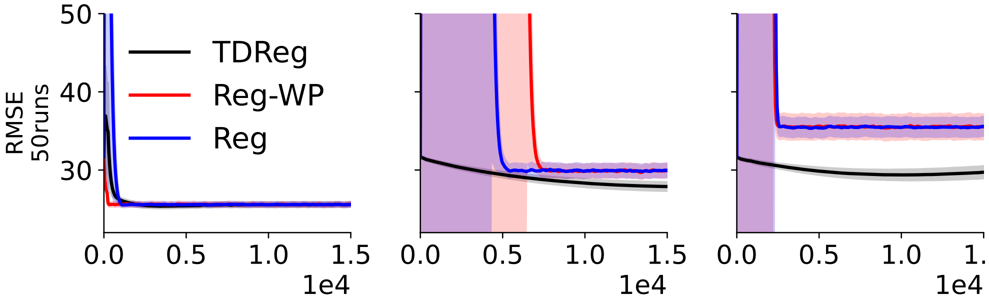
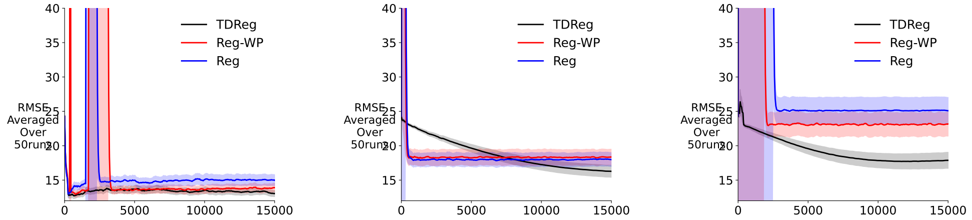


Figure 5: Learning curves on execute time dataset. From left to right, the noise defined as $c \times \epsilon$ where ϵ is noise generated by GP process and $c \in \{10, 20, 30\}$. Due to the addition of large-scale noise, Reg-WP and Reg exhibit high instability during the early learning stages and ultimately converge to a suboptimal solution compared to our TD (in black line) approach.



(a) House, $\sigma = 10$

(b) House, $\sigma = 20$

(c) House, $\sigma = 30$

Figure 6: Learning curves on house dataset. From left to right, the noise variance increases, with $\sigma \in 10, 20, 30$. The results are averaged over 50 runs. Due to the addition of large-scale noise, the two baseline algorithms, Reg-WP and Reg, exhibit high instability during the early learning stages and ultimately converge to a suboptimal solution compared to our TD approach.

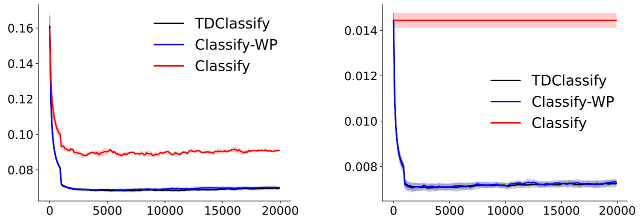
Recall that we employ three intuitive types of transition matrices: $P(\mathbf{x}'|\mathbf{x})$ is larger when the two points $\mathbf{x}, \mathbf{x}'$ are 1) similar (denoted as P_s); 2) far apart (P_f); 3) $P(\mathbf{x}'|\mathbf{x}) = 1/n, \forall \mathbf{x}, \mathbf{x}' \in \mathcal{X}$ (P_c).

Since our results in Figure 8 indicate that $\mathbf{P}$ does not significantly impact regular regression, we conducted experiments on binary classification tasks and observed their particular utility when dealing with imbalanced labels. We define P_s by defining the probability of transitioning to the same class as 0.9 and to the other class as 0.1. Table 3 presents the reweighted balanced results for three binary classification datasets with class imbalance. It is worth noting that in such cases, Classify-WP serves as both 1) no bootstrap baseline and 2) the upsampling techniques for addressing class imbalance in the literature (Kubát & Matwin, 1997).

Observing that TD-Classify and Classify-WP yield nearly identical results and Classify (without using TD's sampling) is significantly worse, suggesting that the benefit of TD arises from the sampling distribution rather than the bootstrap estimate in the imbalanced case. Furthermore, P_f, P_s yield almost the same results in this scenario since they provide the same stationary distribution (equal weight to each class), so here Classify-WP represents both. We also conducted tests using P_c, which yielded results that are almost the same as Classify, and have been omitted from the table. In conclusion, the performance difference of TD in the imbalanced case arises from the transition probability matrix rather than the bootstrap target. The transition matrix's impact is due to the implied difference in the stationary distribution.

Algs Dataset	**TD-Classify**	**Classify-WP**	**Classify**	**TD-WOF**
Insurance	0.0073 ± 0.0001	0.0073 ± 0.0001	0.0144 ± 0.0003	0.4994 ± 0.0005
Weather	0.0695 ± 0.0008	0.0701 ± 0.0008	0.0913 ± 0.0010	0.4965 ± 0.0031

Table 3: Binary classification with imbalance. 0.0073 means 0.73% misclassification rate. The results are smoothed over 5 evaluations before averaging over 5 random seeds.



(a) weather (b) insurance

Figure 7: Learning curves of binary classification with imbalance: balanced testing error v.s. training steps. The results have been smoothed using a 5-point window before averaging over 5 runs.

Alg. / Data	**TD-Reg**	**Reg-WP**	**Reg**
house	3.384 ± 0.21	3.355 ± 0.23	3.319 ± 0.16
exectime	23.763 ± 0.38	23.794 ± 0.36	23.87 ± 0.36
bikeshare	40.656 ± 0.77	40.904 ± 0.36	40.497 ± 0.45

Table 4: Test root mean squared error (RMSE) with standard error. The results have been smoothed using a 5-point window before averaging over 5 runs.

The usage of inverse link function. The results of TD on classification without using a transformation/link function are presented in Table 3 and are marked by the suffix 'WOF.' These results are not surprising, as the bootstrap estimate can potentially disrupt the TD target entirely. Consider a simple example where a training example $\mathbf{x}$ has a label of one and transitions to another example, also labeled one. Then the reward ($r = y - \gamma y'$) will be $1 - \gamma$. If the bootstrap estimate is negative, the TD target might become close to zero or even negative, contradicting the original training label of one significantly.

On binary classification dataset, weather and insurance, the imbalance ratios (proportion of zeros) are around $72.45\%, 88.86\%$ respectively. We set number of iterations to 20k for both and it is sufficient to see convergence. Additionally, in our TD algorithm, to prevent issues with inverse link functions, we add or subtract 1×10^{-15} when applying them to values of 0 or 1 in classification tasks. It should be noted that this parameter can result in invalid values depending on concrete loss implementation, usually setting $< 10^{-7}$ should be generally good.

On those class-imbalanced datasets, when computing the reweighted testing error, we use the function from `https://scikit-learn.org/stable/modules/generated/sklearn.metrics.balanced_accuracy_score.html`.

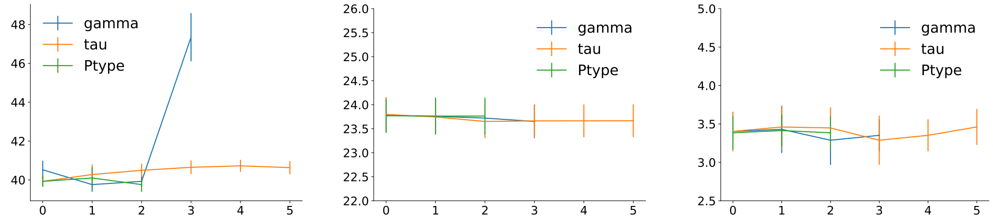
C.5 Deep Learning: Standard Problems

This section aims at investigating how our TD algorithm works on commonly used datasets where there may no be correlation among targets. We found that there is no clear gain/lose with TD's bootstrap target in either regression (Table 4) or image classification tasks (Table 5). For image datasets, we employed a convolutional neural network (CNN) architecture and ResNN18 (He et al., 2016) to demonstrate TD's practical utility.

We then further study the hyperparameter sensitivity in Figure 8. With NNs, TD can pose challenges due to the interplay of two additional hyperparameters: γ and the target NN moving rate τ (Mnih et al., 2015). Optimizing these hyperparameters can often be computationally expensive. Therefore, we investigate their impact on performance by varying γ, τ. We find that selecting appropriate parameters tends to be relatively straightforward. Figure 8 displays the testing performance across various parameter settings. It is evident that performance does not vary significantly across settings, indicating that only one hyperparameter or a very small range of hyperparameters needs to be considered during the hyperparameter sweep. It should be noted that Figure 8 also illustrates the sensitivity analysis when employing three intuitive types of transition matrices that do not require

Algs Dataset	**TD-Classify**	**Classify**
mnist	99.06%	99.00%
mnistfashion	89.10%	88.96%
cifar10	67.42%	67.13%
cifar100	31.55%	31.21%
cifar10(Resnn18)	84.28%	84.37%
cifar100(Resnn18)	56.42%	56.36%

Table 5: Image classification test accuracy. The results are smoothed over 10 evaluations before averaging over 3 random seeds. The standard errors are small with no definitive advantage observed for either algorithm.



(a) Bikeshare

(b) ExecutionTime

(c) House

Figure 8: Sensitivity to hyperparameter settings. We show test RMSE with error bars while sweeping over discount rate $\gamma \in \{0.1, 0.2, 0.4, 0.9\}$, target NN moving rate $\tau \in \{0.001, 0.01, 0.1, 0.2, 0.4, 0.9\}$, and three types of transition probability. γ hurts when it goes to the largest value on bikeshare, and it is likely because the exponential term in Poisson regression needs a unusually small learning rate. When generating curves for τ and γ, we maintain P_{type} as a simple uniform constant.

prior knowledge. It appears that there is no clear gain/lose with these choices in a standard task. We refer readers to Appendix C.7 for details.

C.6 Intuition on Air Quality dataset

Although it is impossible to determine the underlying ground truth of the noise structure precisely, we propose the following intuition for potential positive correlations among data points in this dataset.

If air quality measurements are taken at regular intervals (e.g., hourly or daily) at the same location, temporal dependencies might arise between consecutive measurements due to factors such as diurnal variations, weather patterns, or pollution sources. Consequently, the CO concentration measured at one time point may be positively correlated with that measured at adjacent times. Additionally, spatial dependencies could occur if measurements are taken at nearby locations, particularly in densely populated urban areas with unevenly distributed pollution sources. In such cases, neighboring locations may experience similar air quality conditions, resulting in a positive correlation between their CO concentrations.

C.7 On the Implementation of P

As we mentioned in Section 6, to investigate the effect of transition matrix, we implemented three types of transition matrices: $\mathbf{P}(\mathbf{x}'|\mathbf{x})$ is larger when the two points x, x' 1) are similar; 2) when they are far apart; 3) $\mathbf{P}(\mathbf{x}'|\mathbf{x}) = 1/n, \forall \mathbf{x}, \mathbf{x}' \in \mathcal{X}$. To expedite computations, P_s, P_f are computed based on the training targets instead of the features. The rationale for choosing these options is as follows: the first two may lead to a reduction in the variance of the bootstrap estimate if two consecutive points are positively or negatively correlated.

The resulting matrix may not be a valid stochastic matrix, we use DSM projection (Wang et al., 2010) to turn it into a valid one.

We now describe the implementation details. For first choice, given two points $(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2)$, the formulae to calculate the similarity is:

$$k(y_1, y_2) = \exp(-(y_1 - y_2)^2/v) + 0.1 \tag{73}$$

where v is the variance of all training targets divided by training set size. The second choice is simply $1 - \exp(-(y_1 - y_2)^2/v)$.

Note that the constructed matrix may not be a valid probability transition matrix. To turn it into a valid stochastic matrix, we want: $\mathbf{P}$ itself must be row-normalized (i.e., $\mathbf{P1} = \mathbf{1}$). To ensure fair comparison, we want equiprobable visitations for all nodes/points, that is, the stationary distribution is assumed to be uniform: $\boldsymbol{\pi} = \frac{1}{n}\mathbf{1}$.

The following proposition shows the necessary and sufficient conditions of the uniform stationary distribution property:[2]

Proposition C.1. $\boldsymbol{\pi} = \frac{1}{n}\mathbf{1}$ *is the stationary distribution of an ergodic* $\mathbf{P}$ *if and only if* $\mathbf{P}$ *is a doubly stochastic matrix (DSM).*

Proof. **If**: Note that

$$\pi_j = \sum_i \pi_i p_{ij} \quad \text{and} \quad 1 = \sum_i p_{ij} \tag{74}$$

Subtracting these two gives

$$1 - \pi_j = \sum_i (1 - \pi_i) p_{ij} \quad \text{and} \quad \frac{1 - \pi_j}{n} = \sum_i \frac{1 - \pi_i}{n} p_{ij}.$$

This last equation indicates that $\left(\frac{1-\pi_1}{n}, \frac{1-\pi_2}{n}, \cdots, \frac{1-\pi_n}{n}\right)^\top$ is also the stationary distribution of $\mathbf{P}$. Due to the uniqueness of the stationary distribution, we must have

$$\frac{1 - \pi_i}{n} = \pi_i$$

and thus $\pi_i = \frac{1}{n}, \forall i$.

Only if: Since $\boldsymbol{\pi} = \frac{1}{n}\mathbf{1}$ is the stationary distribution of $\mathbf{P}$, we have

$$\frac{1}{n}\mathbf{1}^\top \mathbf{P} = \frac{1}{n}\mathbf{1}^\top$$

and thus $\mathbf{1}^\top \mathbf{P} = \mathbf{1}^\top$ showing that $\mathbf{P}$ is also column-normalized. □

With linear function approximation,

$$\mathbf{A} = \mathbf{X}^\top \mathbf{D}(\mathbf{I} - \gamma\lambda\mathbf{P})^{-1}(\mathbf{I} - \gamma\mathbf{P})\mathbf{X} \tag{75}$$

$$\mathbf{b} = \mathbf{X}^\top \mathbf{D}(\mathbf{I} - \gamma\lambda\mathbf{P})^{-1}(\mathbf{I} - \gamma\mathbf{P})\mathbf{y} \tag{76}$$

$$\mathbf{A} = \mathbf{X}^\top \mathbf{D}(\mathbf{I} - \gamma\mathbf{P})\mathbf{X} \tag{77}$$

$$\mathbf{b} = \mathbf{X}^\top \mathbf{D}(\mathbf{I} - \gamma\mathbf{P})\mathbf{y} \tag{78}$$

where $\mathbf{X} \in \mathbb{R}^{n \times d}$ is the feature matrix, and $\mathbf{D}$ is uniform when $\mathbf{P}$ is a DSM.

As a result, we can apply a DSM (Wang et al., 2010) projection method to our similarity matrix.

[2] We found a statement on Wikipedia (`https://en.wikipedia.org/wiki/Doubly_stochastic_matrix`), but we are not aware of any formal work that formally supports this. If such support exists, please inform us, and we will cite it accordingly.