Received: 20 December 2022 | Revised: 24 February 2023 | Accepted: 11 March 2023

DOI: 10.1002/sta4.564

ORIGINAL ARTICLE

WILEY

Deep spectral Q-learning with application to mobile health

Yuhe Gao[1] | Chengchun Shi[2] | Rui Song[1]

[1]Department of Statistics, North Carolina State University, Raleigh 27695, USA

[2]Department of Statistics, London School of Economics and Political Science, London WC2A 2AE, UK

Correspondence
Rui Song, Department of Statistics, North Carolina State University, Raleigh, NC 27695, USA.
Email: rsong@ncsu.edu

Dynamic treatment regimes assign personalized treatments to patients sequentially over time based on their baseline information and time-varying covariates. In mobile health applications, these covariates are typically collected at different frequencies over a long time horizon. In this paper, we propose a deep spectral Q-learning algorithm, which integrates principal component analysis (PCA) with deep Q-learning to handle the mixed frequency data. In theory, we prove that the mean return under the estimated optimal policy converges to that under the optimal one and establish its rate of convergence. The usefulness of our proposal is further illustrated via simulations and an application to a diabetes dataset.

KEYWORDS
dynamic treatment regimes, mixed frequency data, principal component analysis, reinforcement learning

1 | INTRODUCTION

Precision medicine focuses on providing personalized treatment to patients by taking their personal information into consideration (see, e.g., Kosorok & Laber, 2019; Tsiatis et al., 2019). It has found various applications in numerous studies, ranging from the cardiovascular disease study to cancer treatment and gene therapy (Jameson & Longo, 2015). A dynamic treatment regime (DTR) consists of a sequence of treatment decisions rules tailored to each individual patient's status at each time, mathematically formulating the idea behind precision medicine. One of the major objectives in precision medicine is to identify the optimal dynamic treatment regime that yields the most favorable outcome on average.

With the rapidly development of mobile health (mHealth) technology, it becomes feasible to collect rich longitudinal data through mobile apps in medical studies. A motivating data example is given by the OhioT1DM dataset (Marling & Bunescu, 2020), which contains data from 12 patients suffering from type-I diabetes measured via fitness bands over 8 weeks. Data-driven decision rules estimated from these data have the potential to improve these patients' health (see, e.g., Shi et al., 2020; Zhu et al., 2020; Zhou et al., 2022). However, it remains challenging to estimate the optimal DTR in these mHealth studies. First, the number of treatment stages (e.g., horizon) is no longer fixed, whereas the number of patients can be limited. For instance, in the OhioT1DM dataset, only 12 patients are enrolled in the study. Nonetheless, suppose treatment decisions are made on an hourly basis, the horizon is over 1000. Existing proposals in the DTR literature (Ertefaie et al., 2021; Fang et al., 2022; Guan et al., 2020; Mo et al., 2021; Murphy, 2003; Nie et al., 2021; Qi & Liu, 2018; Shi et al., 2018; Song et al., 2015; Zhang et al., 2013, 2018; Zhao et al., 2015) become inefficient in these long or infinite horizon settings and require a large number of patients to be consistent. Second, patients' time-varying covariates typically contain mixed frequency data. In the OhioT1DM dataset, some of the variables, such as the continuous glucose monitoring (CGM) blood glucose levels, are recorded every 5 min. Meanwhile, other variables, such as the carbohydrate estimate for the meal and the

Abbreviations: CGM, continuous glucose monitoring; DTR, dynamic treatment regime; IGC, index of glycemic control; mHealth, mobile health; PCA, principal component analysis; ReLU, rectified linear unit; RL, reinforcement learning.

exercise intensity, are recorded with a much lower frequency. Concatenating these high-frequency variables over each 1-h interval produces a high-dimensional state vector, and directly using these states as input of the treatment policy would yield very noisy decision rules. A naive approach is to use some ad hoc summaries of the high-frequency data for policy learning. However, this might produce a suboptimal policy due to the information loss.

Recently, there is a growing line of research in the statistics literature for policy learning and/or evaluation in infinite horizons. Some references include Chen et al. (2022), Ertefaie and Strawderman (2018), Liao et al. (2020), Liao et al. (2021), Li et al. (2022), Luckett et al. (2020), Ramprasad et al. (2022), Shi et al. (2022, 2021), and Xu et al. (2020). In the computer science literature, there is a huge literature on developing reinforcement learning (RL) algorithms in infinite horizons. These algorithms can be casted into as model-free and model-based algorithms. Popular model-free RL methods include value-based methods that model the expected return starting from a given state (or state-action pair) and compute the optimal policy as the greedy policy with respect to the value function (Dabney et al., 2018; Ernst et al., 2005; Mnih et al., 2015; Riedmiller, 2005; Van Hasselt et al., 2016), and policy-based methods that directly search the optimal policy among a parameterized class via policy gradient or actor-critic methods (Koutnk et al., 2013; Mnih et al., 2016; Schulman et al., 2015; Schulman et al., 2015; Wang et al., 2017). Model-based algorithms are different from the model-free algorithms in the sense that they model the transition dynamics of the environment and use the model of environment to derive or improve policies. Popular model-based RL methods include Guestrin et al. (2002), Janner et al. (2019), Lai et al. (2020), and Li et al. (2020), to name a few. Recently, Chen et al. (2021), Janner et al. (2021), and Villaflor et al. (2022) have formulated RL as a sequence modeling problem, where the dynamics of state-action-reward is captured by some transformer architectures such as GPT (Radford et al., 2018). See also Arulkumaran et al. (2017), Luo et al. (2022), Sutton and Barto (2018), and Villaflor et al. (2022) for more details. These methods cannot be directly applied to datasets such as OhioT1DM as they haven't considered the mixed frequency data.

In the RL literature, a few works have considered dimension reduction to handle the high dimensional state system. In particular, Murao and Kitamura (1997) proposed to segment the state space and learn a cluster representation of the states. Whiteson et al. (2007) proposed to divide the state space into tilings to represent each state. Both papers proposed to discretize the state space for dimension reduction. However, it can lead to considerable information loss (Wang et al., 2017). Sprague (2007) proposed an iterative dimension reduction method using neighborhood components analysis. Their method uses a linear basis function to model the Q-function and cannot allow more general nonlinear function approximation. Recently, there are a few works that employ principal components analysis (PCA) for dimension reduction in RL (Curran et al., 2015, 2016; Parisi et al., 2017). However, none of the aforementioned papers formally established the theoretical guarantees for their proposals. Moreover, these methods are motivated by applications in games or robotics, and their generalization to mHealth applications with mixed frequency data remains unknown.

In the DTR literature, a few works considered mixed frequency data, which include both scalar and functional covariates. Specifically, McKeague and Qian (2014) proposed a functional regression model for optimal decision making with one functional covariate. Ciarleglio et al. (2015) and Ciarleglio et al. (2016) extended their proposal to a more general setting with multiple scalar and functional covariates. Ciarleglio et al. (2018) considered variable selection to handle the mixed frequency data. Laber and Staicu (2018) applied functional PCA to the functional covariates for dimension reduction. All these works considered single-stage decision making. Their methods are not directly applicable to the infinite horizon settings.

Our contributions are as follows. Scientifically, mixed frequency data frequently arise in mHealth applications. Nonetheless, it has been less explored in the infinite horizon settings. Our proposal thus fills a crucial gap and greatly extends the scope of existing approaches to learning DTRs. Methodologically, we propose a deep spectral Q-learning algorithm for dimension reduction. The proposed algorithm achieves a better empirical performance than those that either directly use the original mixed frequency data or its ad hoc summaries as input of the treatment policy. Theoretically, we derive an upper error bound for the regret of the proposed policy and decompose this bound into the sum of approximation error and estimation error. Our theories offer some general guidelines for practitioners to select the number of principal components in the proposed algorithm.

The rest of this paper is organized as follows. We introduce some background about DTR and the mixed frequency data in Section 2. We introduce the proposed method to estimate the optimal DTR in Section 3 and study its theoretical properties in Section 4. Empirical studies are conducted in Section 5. In Section 5.3, we apply the proposed method to the OhioT1DM dataset. Finally, we conclude our paper in Section 6.

2 | PRELIMINARY

2.1 | Data and notations

Suppose the study enrolls N patients. The dataset for the i-th patient can be summarized as $O_i \equiv \{(S_{i,t}, A_{i,t}, R_{i,t}) : 1 \le t \le T_i\}$. For simplicity, we assume $T_i = T$ for any i. Each state $S_{i,t} = (X_{i,t}, \{Z_{i,t,j}\}_{j=1}^{J}) \in \mathcal{S}$ is composed of a set of low-frequency covariates $X_{i,t} \in \mathbb{R}^{m_0}$ and a set of J high-frequency covariates $\{Z_{i,t,j}\}_{j=1}^{J}$, where $\mathcal{S}$ is the state space. For each high-frequency variable $Z_{i,t,j}, j \in \{1,2,..,J\}$, we have $Z_{i,t,j} =$

$(Z_{i,t,j}^{(1)}, Z_{i,t,j}^{(2)}, \ldots, Z_{i,t,j}^{(m_j)})^T \in \mathbb{R}^{m_j}$ for some large integer m_j. Let τ denote the length of a time unit. The low-frequency variables are recorded at time points $\tau, 2\tau, \ldots, t\tau, \ldots$. The jth high-frequency variables, however, are recorded more frequently at time points $m_j^{-1}\tau, 2m_j^{-1}\tau, \ldots$. Notice that we allow the J high-frequency variables to be recorded with different frequencies. Let $m = \sum_{j=1}^{J} m_j$ and $Z_{i,t}^T = [Z_{i,t,1}^T, Z_{i,t,2}^T, \ldots, Z_{i,t,J}^T] \in \mathbb{R}^m$ denote a high-dimensional variable that concatenates all the high-frequency covariates. As such, the state $S_{i,t}$ can be represented as $(X_{i,t}, Z_{i,t}) \in \mathbb{R}^{m_0+m}$. In addition, $A_{i,t}$ denotes the treatment indicator at the tth time point and $\mathcal{A}$ denotes the finite set of all possible treatment options with size $|\mathcal{A}| \in \mathbb{N}$. The reward, $R_{i,t}$ corresponds to the ith patient's response obtained after the tth decision point. By convention, we assume a larger value of $R_{i,t}$ indicates a better outcome. We require $|R_{i,t}|$ to be uniformly bounded by some constant $R_{max} > 0$, and assume $O_1, O_2, \ldots, O_N$ are *i.i.d.*, which are commonly imposed in the RL literature (see, e.g., Sutton & Barto, 2018). Finally, we denote the l_p-norm of a function aggregated over a given distribution function σ by $\|.\|_{p,\sigma}$. We use $[q]$ to represent the indices set $\{1,2,3,\ldots q\}$ for any integer $q \in \mathbb{N}$.

2.2 | Assumptions, policies, and value functions

We will require the system to satisfy the Markov assumption such that

$$P(S_{0,t+1} \in \mathbb{S} | S_{0,t} = s, A_{0,t} = a, \{S_{0,t'}, A_{0,t'}\}_{0 \le t' < t}) = P(S_{0,t+1} \in \mathbb{S} | S_{0,t} = s, A_{0,t} = a), \forall t,$$

for any s, a and Borel set $\mathbb{S} \in \mathcal{S}$. In other words, the distribution of the next state depends on the past data history only through the current state-action pair. We assume the transition kernel is absolutely continuous with respect to some uniformly bounded transition density function $q(s'|s,a)$ such that $\sup_{s,a,s'} |q(s'|s,a)| \le c_q$ for some constant $c_q > 0$.

In addition, we also impose the following conditional mean independence assumption:

$$E(R_{0,t} | S_{0,t} = s, A_{0,t} = a, \{R_{0,t'}, S_{0,t'}, A_{0,t'}\}_{0 \le t' < t}) = E(R_{0,t} | S_{0,t} = s, A_{0,t} = a) = r(s,a), \forall t,$$

where we refer to $r(s,a) = r(x, \{z_j\}_{j=1}^J, a)$ as the immediate reward function.

Next, define a policy $\pi : \mathcal{S} \to P(\mathcal{A})$ as a function that maps a patient's state at each time point into a probability distribution function on the action space. Given π, we define its (state) value function as

$$V^{\pi}(s) = \sum_{t=0}^{\infty} \gamma^t E^{\pi}\{R_{0,t} | S_{0,0} = s\},$$

with $\gamma \in [0,1)$ being a discount factor that balances the immediate and long-term rewards. By definition, the state-value function characterizes the expected return assuming the decision process follows a given target policy π. In addition, we define the state-action value function (or Q-function) as

$$Q^{\pi}(s,a) = \sum_{t=0}^{\infty} \gamma^t E^{\pi}\{R_{0,t} | S_{0,0} = s, A_{0,0} = a\},$$

which is the expected discounted cumulative rewards given an initial state-action pair.

Under the Markov assumption and the conditional mean independence assumption, there exists an optimal policy π^* such that $V^{\pi^*}(s) \ge V^{\pi}(s), \forall \pi, s \in \mathcal{S}$ (see, e.g., Puterman, 2014). Moreover, π^* satisfies the following Bellman optimality equation:

$$Q^{\pi^*}(s,a) = E\{R_{0,t} + \gamma \max_{a' \in \mathcal{A}} Q^{\pi^*}(S_{0,t+1}, a') | S_{0,t} = s, A_{0,t} = a\}, \quad (1)$$

where Q^{π^*} denotes the optimal Q-function.

2.3 | Rectified linear unit (ReLU) network

In this paper, we use value-based methods that learn the optimal policy π^* by estimating the optimal Q-function. We will use the class of sparse neural network with the ReLU activation function, i.e., $f(x) = \max(x, 0)$, to model the Q-function. The advantage of using a neural network over a simple parametric model is that the neural network can better capture the potential nonlinearity in the high-dimensional state system.

Formally, the class of sparse ReLU network is defined as

$$\mathcal{F}_{SReLU}(L,\{d_j\}_{j=0}^{L+1},s,V_{max}) = \{\mathbb{R}^{d_0} \to \mathbb{R}^{d_{L+1}} : f(x) = W_L g_{L-1} \circ g_{q-1} \circ \ldots g_1 \circ g_0(x),$$
$$g_j(x) = \sigma(W_j x + v_j), W_j \in \mathbb{R}^{d_{j+1} \times d_j}, v_j \in \mathbb{R}^{d_{j+1}}, j \in \{0,1,\ldots,L\},$$
$$\max_{j=0,1,\ldots,L}\{\max(\|W_j\|_\infty, |v_j|_\infty)\} \le 1, \sum_{j=0}^{L}(\|W_j\|_0 + |v_j|_0) \le s, \max_{k \in \{1,2,\ldots,d_{L+1}\}} \|f_k\|_\infty \le V_{max}\}.$$

Here, L is the number of hidden layers of the neural network and d_j is the width of each layer. The output dimension d_{L+1} is set to 1 since the Q-function output is a scalar. The parameters in $\mathcal{F}_{ReLU}(L)$ are the weight matrices W_j and bias vectors v_j. The sparsity level s upper bounds the total number of nonzero parameters in the model. This constraint can be satisfied using dropout layers in the implementation (Srivastava et al., 2014). In theory, sparse ReLU networks can fit smooth functions with a minimax optimal rate of convergence (Schmidt-Hieber, 2020). The main theorems in Section 4 will rely on this property. An illustration of sparse ReLU network is in Figure 1.

3 | SPECTRAL FITTED Q-ITERATION

Neural network with ReLU activation functions in Section 2.3 is commonly used in value-based reinforcement learning algorithms. However, in medical studies, the training dataset is often of limited size, with a few thousands or tens of thousands of observations in total (see, e.g., Liao et al., 2021; Marling & Bunescu, 2020). Meanwhile, the data contain high-frequency state variables, which yields a high-dimensional state system. Directly using these states as input will procedure a very noisy policy. This motivates us to consider dimension reduction in RL.

A naive approach for dimension reduction is to use some summary statistics of the high-frequency state as input for policy learning. For instance, on the OhioT1DM dataset, the average of CGM blood glucose levels between two treatment decision points can be used as the summary statistic, as in Shi et al. (2022), Zhu et al. (2020), and Zhou et al. (2022). In this paper, we propose to use principal component analysis to reduce the dimensionality of $\{Z_{i,t,j}\}_{j=1}^{J}$. We expect that using PCA can preserve more information than some ad hoc summaries (e.g., average).

To apply PCA in the infinite horizon setting, we need to impose some stationarity assumptions on the concatenated high-dimensional variables $Z_{i,t}^T \in \mathbb{R}^m$: $E[Z_{i,t}] = \boldsymbol{\mu}$ and $Cov[Z_{i,t}] = \mathbf{G}$ for some mean vector $\boldsymbol{\mu} \in \mathbb{R}^m$ and covariance matrix $\mathbf{G} \in \mathbb{R}^{m \times m}$ that are independent of t. In real data application, we can test whether the concatenated high-frequency variable $Z_{i,t}$ is weak stationary (see, e.g., Dickey & Fuller, 1979; Kwiatkowski et al., 1992; Said & Dickey, 1984). If it is weak stationary, the concatenated high-frequency covariate $Z_{i,t}$ will automatically satisfy the two assumptions above. Similar assumptions have been widely imposed in the literature (see, e.g., Kallus & Uehara, 2022; Shi et al., 2021). Without loss of generality, we assume $\boldsymbol{\mu} = 0_m$ for simplicity of notations. For the covariance matrix $\mathbf{G}$, it is generally unknown. In practice, we recommend to use the sample covariance estimator $\hat{\mathbf{G}}$.

We describe our procedure as follows. By the spectral decomposition, we have $\hat{\mathbf{G}} = \sum_{k=1}^{m} \hat{\lambda}_k \hat{U}_k \hat{U}_k^T$, where $\lambda_1 \ge \lambda_2 \ge \ldots \ge 0$ are the eigenvalues and $\hat{U}_k$'s are the corresponding eigenvectors. This allows us to represent $Z_{i,t}$ as $\sum_{k=1}^{m} \hat{\lambda}_k^{1/2} \hat{V}_k^{(i,t)} \hat{U}_k$, where $\hat{V}_k^{(i,t)}$s are the estimated principal component scores, given by $\hat{\lambda}_k^{-1/2} Z_{i,t}^T \hat{U}_k$. For any κ, the estimated principal component scores $\hat{\mathbf{V}}_{i,t,\kappa} = (\hat{V}_1^{(i,t)}, \hat{V}_2^{(i,t)}, \ldots, \hat{V}_\kappa^{(i,t)})$ correspond to the $\kappa \le m$ largest eigenvalues of the concatenated high-frequency variable $Z_{i,t}$. When $\kappa = m$, using these principal component scores is equivalent to using the original high-frequency variable $Z_{i,t}$. We will approximate $Q^\pi(X_{i,t}, \mathbf{V}_{i,t,m}, A_{i,t})$ by $Q^\pi(X_{i,t}, \hat{\mathbf{V}}_{i,t,\kappa}, A_{i,t})$ and propose to use neural fitted Q-iteration algorithm by Riedmiller (2005) to learn the estimated optimal policy. We detail our procedure in Algorithm 1.

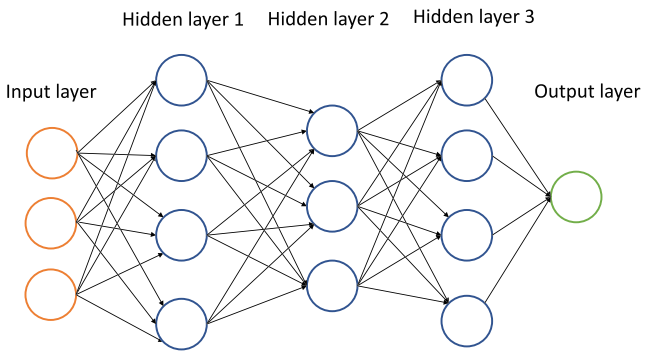

FIGURE 1 Illustration of a rectified linear unit (ReLU) network.

Algorithm 1 Spectral fitted Q-iteration

Input: $\{S_{i,t} = (X_{i,t}, Z_{i,t}), A_{i,t}, R_{i,t}, \gamma\}$ with $i \in \{1, 2, \dots, N\}, t \in \{1, 2, \dots, T\}$; ReLU network function class $\mathcal{F}_{SReLU} = \mathcal{F}_{SReLU}(L, \{d_j\}_{j=0}^{L+1}, s, V_{max})$; sampling distribution σ; sample size n; number of principal components κ; number of iterations K; estimated covariance matrix $\hat{\mathbf{G}}$ for $Z_{i,t}$;

Calculate first κ PCA $\hat{\mathbf{V}}_{i,t,\kappa}$ for high-frequency data part $Z_{i,t}$;

For each $a \in \mathcal{A}$, initialize a sparse ReLU network $\tilde{Q}_0(x, v_\kappa, a) \in \mathcal{F}_{SReLU}$;

for $k = 1$ **to** K **do**

Sample n observations $I_k = \{(i,t) : 1 \le i \le N, 1 \le t \le T-1\}$ based on σ from data;

Define a response $Y_{i,t}$ based on $\tilde{Q}_k$: $Y_{i,t}(\tilde{Q}_k) = R_{i,t} + \gamma \max_{a \in \mathcal{A}} \tilde{Q}_k(X_{i,t+1}, \hat{\mathbf{V}}_{i,t+1,\kappa}, a)$;

Update $\tilde{Q}_k$ to $\tilde{Q}_{k+1}$:

$$\tilde{Q}_{k+1} \leftarrow \underset{f(.,a) \in \mathcal{F}_{SReLU}}{\operatorname{argmin}} \frac{1}{n} \sum_{(i,t) \in I_j} [Y_{i,t}(\tilde{Q}_k) - f(X_{i,t}, \hat{\mathbf{V}}_{i,t,\kappa}, A_{i,t})]^2$$

end for

Return The greedy policy: $\pi_K(a|x, v_\kappa) = 0$, if $a \notin \operatorname{argmax}_{a'} \tilde{Q}_K(x, v_\kappa, a'), \forall x, v_\kappa, a$

In Algorithm 1, we fit $|\mathcal{A}|$ neural networks corresponding to each a in $Q(s,a)$. This is reasonable in settings where the action space is small. For the ReLU network $\tilde{Q}_k(\cdot,a), k = 0,1,\dots,K$ in Algorithm 1, the input is concatenation of the low-frequency part $X \in \mathbb{R}^{m_0}$ and the principal component vector $\hat{\mathbf{V}} \in \mathbb{R}^\kappa$. The input dimension for $\tilde{Q}_k(\cdot,a)$ is then $m_0 + \kappa$. When the dataset is small (such as the OhioT1DM dataset), we recommend to set n to $N(T-1)$ such that all the data transactions (instead of a random subsample) will be used in each iteration.

Similar to the original neural fitted Q-iteration algorithm in Riedmiller (2005), the intuition of this algorithm is also based on the Bellman optimality equation (1). In each step k of Algorithm 1, $\tilde{Q}_k$ estimates Q^{π^*} and the response $Y_{i,t}(\tilde{Q}_k) = R_{i,t} + \gamma \max_{a \in \mathcal{A}} \tilde{Q}_k(X_{i,t+1}, \hat{\mathbf{V}}_{i,t+1,\kappa}, a)$ corresponds to the right-hand side of Equation (1). Therefore, fitting the regression of $Y_{i,t}$ with $\tilde{Q}_{k+1}$ is to solve the Bellman optimality equation. The key difference between Algorithm 1 and the original neural fitted Q-iteration algorithm is that the high-dimensional input $Z_{i,t}^T = [Z_{i,t,1}^T, Z_{i,t,2}^T, \dots, Z_{i,t,J}^T]$ is involved in the state space and is mapped to a lower dimensional vector $\hat{\mathbf{V}}_{i,t,\kappa}$ during the learning process, so the neural network $\tilde{Q}_0(x, v_\kappa, a)$ takes principle component $\hat{\mathbf{V}}_{i,t,\kappa}$ rather than original high dimensional $Z_{i,t}$ as input.

4 | ASYMPTOTIC PROPERTIES

Before discussing the asymptotic properties of our proposed Q-learning method, we introduce some notations.

Definition 1. Define

$$\mathcal{F}_0(L, \{d_i\}_{i=0}^{L+1}, s, V_{max}) = \{f : \mathcal{S} \times \mathcal{A} \to \mathbb{R}, f(.,a) \in \mathcal{F}_{SReLU}(L, \{d_i\}_{i=0}^{L+1}, s, V_{max}), \forall a \in \mathcal{A}\},$$

where $\mathcal{F}_{SReLU}(L, \{d_j\}_{i=0}^{L+1}, s, V_{max})$ is the class of sparse ReLU network with L layers and sparsity parameter s and $V_{max} = \frac{R_{max}}{1-\gamma}$, the uniform upper bound for the cumulative reward.

We define $L_{\mathcal{F}_0} = \sup_{f \in \mathcal{F}_0} \sup_{x \neq y} \frac{|f(y) - f(x)|^2}{\|y - x\|^2}$ as the Lipschitz constant for the sparse ReLU class $\mathcal{F}_{SReLU}(L, \{d_j\}_{i=0}^{L+1}, s, V_{max})$ used in $\mathcal{F}_0$. Note that this class $\mathcal{F}_0$ is used in the original neural fitted Q-iteration algorithm to model the Q-function, where the dimension of high-frequency part Z in state is not reduced through PCA. We further define a function class $\mathcal{F}_2$ such that it models the Q-function by first converting high-frequency part Z into its principal component scores and then use a sparse ReLU neural network to obtain the resulting Q-function. More specifically, $\mathcal{F}_2$ is a set of functions $\{f_2\}$, such that $f_2(x,z,a) = f_0(x, \hat{v}_\kappa, a)$, where $f_0 \in \mathcal{F}_0$ and $\hat{v}_\kappa$ is the vector containing first κ principle components of Z. Note that $\mathcal{F}_2$ is the function class that we use in Algorithm 1 to model the Q-function. That is, $\tilde{Q}_k \in \mathcal{F}_2, k \in \{1,2,\dots,K\}$ (formal definition of $\mathcal{F}_2$ and another function class $\mathcal{F}_1$ not mentioned here are in Section A of Appendix S1).

In addition, we introduce the Hölder smooth function class by $\mathcal{C}_r(\mathcal{D}, \beta, H)$ with $\mathcal{D} \in \mathbb{R}^r$ to be the set of function input. The definition is as follows:

Definition 2. Define

$$\mathcal{C}_r(\mathcal{D}, \beta, H) = \left\{ f : \mathcal{D} \to \mathbb{R} : \sum_{\gamma < \beta} \|\partial^\gamma f\|_\infty + \sum_{\alpha \in \mathbb{N}_0^r : \|\alpha\| = \lfloor \beta \rfloor} \sup_{x,y \in \mathcal{D}, x \neq y} \frac{|\partial^\alpha f(x) - \partial^\alpha f(y)|}{\|x - y\|_\infty^{\beta - \lfloor \beta \rfloor}} \le H \right\},$$

where $\alpha \in \mathbb{N}_0^r$ is a r-tuple multi-index for partial derivatives.

We next construct a q-layer network structure $\mathcal{G}(\{p_j,t_j,\beta_j,H_j\}_{j\in[q]})$ with the component function on each layer of this network belonging to Holder smooth function class $\mathcal{C}(\mathcal{D},\beta,\mathcal{H})$, which is called composition of Holder Smooth functions. This composition network contains q layers, with each layer being $g_j:[a_j,b_j]^{p_j}\to[a_{j+1},b_{j+1}]^{p_{j+1}}$, such that g_{jk} the kth component ($k\in[p_{j+1}]$) in layer j satisfies that $g_{jk}\in\mathcal{C}_{t_j}([a_j,b_j]^{t_j},\beta_j,H_j)$ with $1\le t_j\le p_j$ inputs. Similar to the definition of $\mathcal{F}_0$, we can define the function class $\mathcal{G}_0(\{p_j,t_j,\beta_j,H_j\}_{j\in[q]})$ (simply denoted as $\mathcal{G}_0$) on $\mathcal{S}\times\mathcal{A}\to\mathbb{R}$ such that each function $g\in\mathcal{G}_0$ satisfies that $g(.,a)\in\mathcal{G}(\{p_j,t_j,\beta_j,H_j\}_{j\in[q]})$ for $\forall a\in\mathcal{A}$. The relation between function class $\mathcal{G}_0$ and the network structure $\mathcal{G}$ is similar to the relation between function class $\mathcal{F}_0$ and the neural network $\mathcal{F}_{SReLU}$ in Definition 1. See Definition 4.1 of Fan et al. (2020) for more details on $\mathcal{G}_0(\{p_j,t_j,\beta_j,H_j\}_{j\in[q]})$.

Next, we will introduce the three major assumptions for our theorems:

Assumption 1. The eigenvalues of $Cov(Z)$ follow an exponential decaying trend $\lambda_k=O(e^{-\zeta k}),k=1,2,\ldots,m$ for some constant $\zeta>0$.

Assumption 2. The estimator $\hat{\mathbf{G}}=\sum_{k=1}^m\hat{\lambda}_k\hat{U}_k\hat{U}_k^T$ satisfies that $\left\|\hat{U}_k-U_k\right\|_2=O_p(n^{-\Delta})$ for $1\le k\le m$ such that $\Delta>0$ is some constant.

Assumption 3. First, we define the Bellman optimality operator $\mathcal{T}$ as

$$\mathcal{T}f(x,z,a)=E\left\{R_{0,t}+\gamma\max_{a'\in\mathcal{A}}f(X_{0,t+1},Z_{0,t+1},a')|A_{0,t}=a,X_{0,t}=x,Z_{0,t}=z\right\}.$$

Then we assume $\mathcal{T}f\in\mathcal{G}_0$ for $f\in\mathcal{F}_2$.

Among the three assumptions, the exponential decaying structure of eigenvalues in Assumption 1 can be commonly found in the literature of high-dimensional and functional data analysis (see, e.g., Crambes & Mas, 2013; Jirak, 2016; Reiß & Wahl, 2020). This assumption is to control the information loss caused by using the first κ principal component scores of $Z_{i,t}$ only. Assumption 2 is about the consistency of the estimators $\hat{\mathbf{G}}$ and similar assumptions are imposed in the literature of functional data analysis (see, e.g., Laber & Staicu, 2018; Staicu et al., 2014). Using similar arguments in proving Theorem 5.2 of Zhang and Wang (2016), we can show that such an assumption holds in our setting as well. It is to bound the error caused by the estimation of the covariance matrix. Assumption 3 is referred to as the completeness assumption in the literature (see, e.g., Chen & Jiang, 2019; Uehara et al., 2021, 2022). This assumption is automatically satisfied when the transition kernel and the reward function satisfy certain smoothness conditions.

Our first theorem is concerned with the convergence rate of $\tilde{Q}_K$ in Algorithm 1.

Theorem 1 Convergence of estimated Q-function. *Let μ be some distribution on $\mathcal{S}$ such that $\frac{d\mu(s)}{ds}$ is bounded away from 0. Under Assumptions 1 to 3, with sufficiently large n, there exists a sparse ReLU network structure for the function class $\mathcal{F}_2$ modeling $\tilde{Q}(s,a)$, such that $\tilde{Q}_K$ obtained from our Algorithm 1 satisfies that*

$$\frac{1}{|\mathcal{A}|}\sum_{a\in\mathcal{A}}\left\|Q^{\pi^*}(.,a)-\tilde{Q}_K(.,a)\right\|_{2,\mu}^2=O_p\left(|\mathcal{A}|(n^{\alpha^*}\log^{\xi^*}n+d_1^*\kappa)n^{-1}\log^{\xi^*+1}n+L_{\mathcal{F}_0}(e^{-\zeta\kappa}-e^{-\zeta m}+n^{-2\Delta})+\frac{\gamma^{2K}}{(1-\gamma)^2}R_{max}^2\right),$$

where $0<\alpha^*<1$ is a constant, $\xi^*>1$ is a coefficient related to the growing rate of neural network layer number with n, $|\mathcal{A}|$ is number of treatment options, and d_1^* is the width of the first layer of the sparse ReLU network used in $\mathcal{F}_2$ satisfying the bound $m_0+m\le d_1^*\le n^{\alpha^*}$.

More details of neural network structure, sample size assumptions, and values of α^*,ξ^* for Theorem 1 can be found in Section B of Appendix S1. Theorem 1 provides an error bound on the estimated Q-function $\tilde{Q}_K$. Based on this theorem, we further establish the regret bound of the estimated policy π_K obtained via Algorithm 1. Toward that end, we need another assumption:

Assumption 4. Assume there exist $\eta>0,\delta_0>0$, such that

$$P\left(s:\max_a Q^{\pi^*}(s,a)-\max_{a\in\mathcal{A}-\text{argmax}_{a'}Q^{\pi^*}(x,a')}Q^{\pi^*}(s,a)\le\epsilon\right)=O(\epsilon^\eta)$$

for $0<\epsilon\le\delta_0$.

The margin type condition Assumption 4 is commonly used in the literature. Specifically, in classification, the margin conditions are imposed to bound the excess risk (Audibert & Tsybakov, 2007; Tsybakov, 2004). In dynamic treatment regime, a similar assumption is introduced for proving the convergence of state-value function in a finite horizon setting (Luedtke & van der Laan, 2016; Qian & Murphy, 2011). In RL, these assumptions were introduced by Shi et al. (2022) to obtain sharper regret bound for the estimated optimal policy.

Theorem 2 Convergence of state-value function. *Under Assumptions 1 to 4 and the conditions of $\mu, \mathcal{F}_2, n$ in Theorem 1, we have*

$$E_\mu[V^{\pi^*}(s) - V^{\pi_K}(s)] = O_p\left(\frac{1}{1-\gamma}\left\{|\mathcal{A}|(n^{\alpha^*}\log^{\xi^*} n + d_1^* \kappa)n^{-1}\log^{\xi^*+1} n + L_{\mathcal{F}_0}(e^{-\zeta\kappa} - e^{-\zeta m} + n^{-2\Delta}) + \frac{\gamma^{2K}}{(1-\gamma)^2}R_{max}^2\right\}^{\frac{\eta+1}{\eta+2}}\right).$$

The proofs of the two theorems are included in Section C of Appendix S1. We summarize our theoretical findings below. First, we notice that the convergence rate of regret in Theorem 2 is faster than the convergence of estimated Q-functions in Theorem 1. This is due to the margin type Assumption 4, which enables us to obtain a sharper error bound. Similar results have been established in the literature. See, for example, Theorem 3.3 in Audibert and Tsybakov (2007), Theorem 3.1 in Qian and Murphy (2011), and Theorems 3 and 4 in Shi et al. (2022). Without Assumption 4, it is equivalent to the case of $\eta = 0$ in Theorem 2, where $\frac{\eta+1}{\eta+2} = \frac{1}{2}$ and the convergence rates of the state-value function and Q-function will be the same.

The regret bound in the theorems is mainly determined by four parameters: The sample size n, the number of principal components κ, the number of iterations K, and the number of layers in the neural network (denoted as L). Here, the first term $|\mathcal{A}|(n^{\alpha^*}(\log^{\xi^*} n) + d_1^*\kappa)n^{-1}\log^{\xi^*+1} n$ on the right-hand side of Theorem 2 corresponds to the estimation error, which decreases with n and increases with κ. The second term $L_{\mathcal{F}_0}(e^{-\zeta\kappa} - e^{-\zeta m} + n^{-2\Delta})$ corresponds to the approximation error, which decreases with both n and κ. The remaining term $\frac{\gamma^{K+1}}{(1-\gamma)^2}R_{max}$ is the optimization error that will decrease as the iteration number K in Algorithm 1 grows. For L, it satisfies $L = C_L \log^{\xi^*} n$ for some constant $C_L > 0$ (details can be found in Section B of Appendix S1). Note that L cannot be too small as it grows faster than log$n$. Furthermore, the error bound of Q-function and state-value function contains a quadratic form of L, as is shown in the term $|\mathcal{A}|(n^{\alpha^*}\log^{\xi^*} n + d_1^*\kappa)n^{-1}\log^{\xi^*+1} n = \frac{|\mathcal{A}|\log n}{C_L^2 n}(n^{\alpha^*}L + C_L d_1^*\kappa)L$ of Theorems 1 and 2. Therefore, larger values of L tend to result in slower convergence.

Compared with the existing results on the convergence rate of deep fitted Q-iteration algorithm (see Theorem 4.4 of Fan et al., 2020), our theorems additionally characterize the dependence upon the number of principal components. Specifically, selecting the first κ principal components induces the information loss (e.g., bias) that is of the order $e^{-\zeta\kappa} - e^{-\zeta m}$ but reduces the model complexity caused by high-frequency variables from $d_1^* m$ to $d_1^*\kappa$ and hence the variance of the policy estimator. This represents a bias-variance trade-off. Notice that the bias decays at an exponential order, when the training data are small, reducing the model complexity can be more beneficial. Thus, our algorithm is expected to perform better than the original fitted Q-iteration algorithm in small samples, as shown in our numerical studies.

Finally, The number of principal components shall diverge with n to ensure the consistency of the proposed algorithm. Based on the two theorems, the optimal κ^* that balances the bias and variance trade-off shall satisfy $\kappa^* \asymp \log(n)$ (details are given in Section D of Appendix S1). Thus, when n goes to infinity, we will eventually take $\kappa^* = m$ and our Algorithm 1 will be equivalent to the original neural fitted Q-iteration. This is just an asymptotic guideline for selecting the number of principal components. We provide some practical guidelines in the next section.

5 | EMPIRICAL STUDIES

5.1 | Practical guidelines for number of principal components

In functional data analysis, several criteria have been developed to select number of principal components, including the percentage of variance explained, Akaike information criterion (AIC), and Bayesian information criterion (BIC) (see, e.g., Li et al., 2013; Yao et al., 2005). In our setting, it is difficult to apply AIC/BIC, since there does not exist a natural objective function (e.g., likelihood) for Q-function estimation. One possibility is to extend the value information criterion (Shi et al., 2021) developed in single-stage decision making to infinite horizons. Nonetheless, it remains unclear how to determine the penalty parameter for consistent tuning parameter selection.

Here, we select κ based on the percentage of variance explained. That is, we can look at the minimum value of κ such that the total variance explained by PCA reaches a certain level (e.g., 95%). This method is also employed in Laber and Staicu (2018) in single-stage decision making. To illustrate the empirical performance of this method, we apply Algorithm 1 with $\kappa \in \{2,6,10,14...,74\}$ and evaluate the expected return of these policies in the following numerical study.

The data generating process can be described as follows. We set the low-frequency covariate $X_{i,t}$ to be a two-dimensional vector and the high-frequency variables $Z_{i,t}$ to be a 108-dimensional vector ($m = 108$). Both are sampled from mean zero normal distributions. The covariance

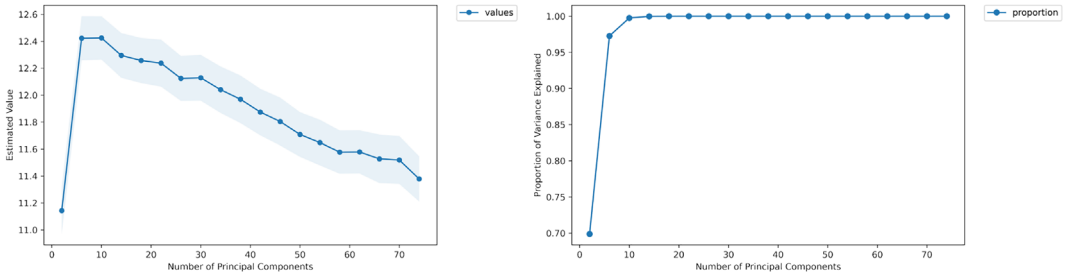

FIGURE 2 Left: Estimated value of policies by Algorithm 1 when κ varies in $\{2,6,10,14...,74\}$ (shaded area is 95% confidence interval); right: proportion of variance explained by the first κ principal components.

matrix of Z is set to satisfy Assumption 1. The action space is binary (i.e., $\mathcal{A}=\{0,1\}$) and the behavior policy to generate actions in training data is a uniform random policy. The reward function $r(x,z,a)$ is set to $=x\beta_{1,a}+z\beta_{2,a}+c\max\{(x\beta_{1,a}+z\beta_{2,a}),0\}$ for some constant c and coefficient vectors $\beta_{1,a},\beta_{2,a}$ such that it is a mixture of a linear function and a neural network with a single layer and ReLU activation function. Next state $(X_{i,t+1},Z_{i,t+1})$ will be generated from a normal distribution with mean being a linear function of state $(X_{i,t},Z_{i,t})$ and action $A_{i,t}$. The number of trajectories N is fixed to 6 and the length of trajectory T is set to be 80.

The ReLU network is constructed with three hidden layers and width $d_1=15,d_2=d_3=5$. Dropout layers with 10% dropout rate are added between layer 1, layer 2, and layer 3. During training, the dropout layers randomly sets the output from previous layers to 0 with the probability 10%, which can introduce sparsity to the neural network and reduce overfitting (Srivastava et al., 2014). The hyperparamters of neural network structure can be tuned via cross-validation. The discounted factor γ is fixed to 0.5.

To evaluate the policy performance, we can use a Monte Carlo method to approximate the expected return under each estimated policy. Specifically, for each estimated policy π, we generate $N_{mc}=100$ trajectories each of length $T_{mc}=20$ (in our setting with $\gamma=0.5$, the cumulative reward after $T_{mc}=20$ is negligible). The initial state distribution is the same as the one in the training dataset. The actions are assigned according to π. The expected return can then be approximated via the average of the empirical cumulative rewards over the 100 trajectories.

For each κ in the list $\{2,6,10,14...,74\}$, we apply Algorithm 1 to learn the optimal policy over 80 random seeds and evaluate their expected return using the Monte Carlo method. We then take the sample average and standard error of these 80 expected returns to estimate the value of policy and construct the margin of error. Figure 2 depicts the estimated values of these expected returns as well as their confidence intervals. It can be seen that increasing κ from 2 to 6 leads to a significant improvement. However, further increasing κ worsens the performance. This trend is consistent with our theory since the bias term will dominate the estimation error for small value of κ. When κ increases, the bias decays at exponentially fast and the model complexity term becomes the leading term. Meanwhile, the percentage of variance explained increases quickly when $\kappa\le 6$ and remains stable afterwards. As such, it makes sense to use this criterion for κ selection. In our implementation, we select the smallest κ such that the variance explained is at least 95%.

5.2 | Simulation study

In the simulation study, we compare the proposed policy π_K^{PCA} against two baseline policies obtained by directly using the original high-frequency variable $Z_{i,t,j}$ (denoted by π_K^{ALL}) and its average as input (denoted by π_K^{AVE}). Both policies are computed in a similar manner based on the deep fitted Q-iteration algorithm. We additionally include one more baseline policy, denoted by π_K^{BOTTLE}, which adds one bottleneck layer after the input of original Z in the neural network architecture such that the width of this bottleneck layer is the same as the input dimension of the proposed policy π_K^{PCA}. This policy differs from the proposed policy in that it uses this bottleneck layer for dimension reduction instead of PCA. Both the data generating setting and the neural network structure are the same to Section 5.1. However, in this simulation study, we vary the sample size and the dimension of the high-frequency variables. Specifically, we consider 15 cases of training size ($N=6,T=30,45,60,75,90,105,120$ and $N=2,3,4,5,7,8,9,10,T=120$) and five different high-frequency part dimension $m=27,54,108,162,216$. In particular, the two cases $N=6,T=30$ and $N=2,T=120$ are corresponding to the scenarios where the size of available training data is extremely small. Furthermore, we have two settings of generating high-frequency variables that will be discussed below. We similarly compare the proposed policy against $\pi_K^{ALL},\pi_K^{BOTTLE}$, and π_K^{AVE} and use the Monte Carlo method to evaluate their values.

In the first setting, we consider five cases with J (the number of high-frequency variables) equal to 1, 2, 4, 6, 8 and each high-frequency variable $Z_j,j\in\{1,2,...,J\}$ is of dimension 27. In this setting, all J high-frequency variables are dependent and eigenvalues of concatenated high-

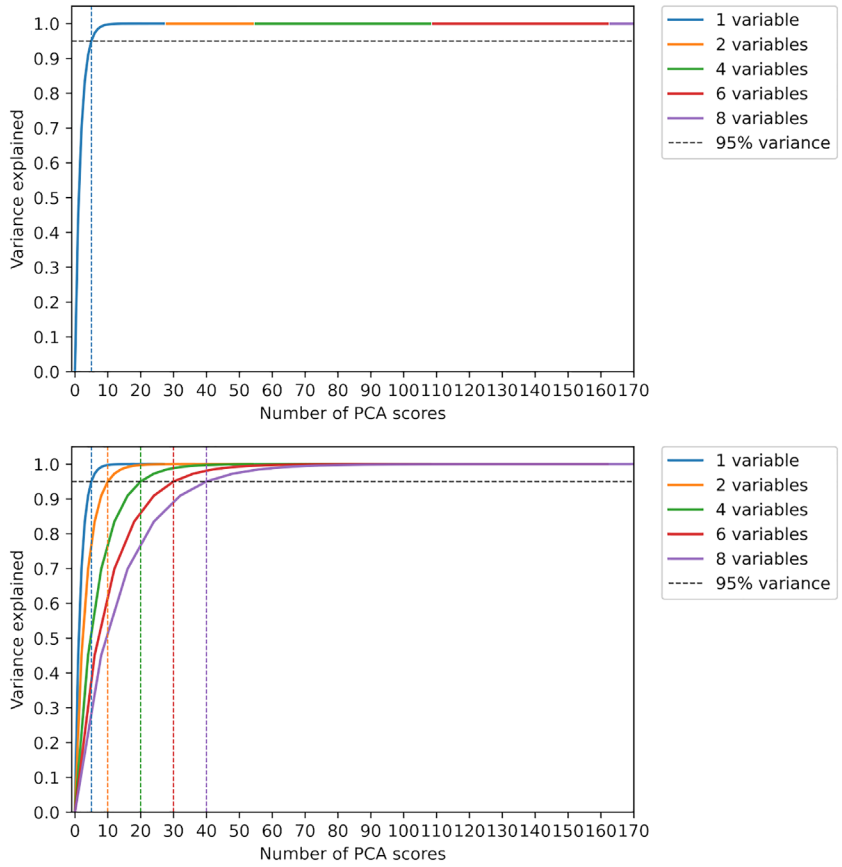

FIGURE 3 Upper: Variance explained by first κ principal component scores when there are $J = 1,2,4,6,8$ high-frequency variables in the **first setting** of simulation (corresponding to Figure 4). Horizontal dash line is 95% of variance explained. $\kappa = 5$ can explain 95% variance in all $J = 1,2,4,6,8$ cases of the first setting; lower: variance explained by first κ principal component scores when there are $J = 1,2,4,6,8$ high-frequency variables in the **second setting** of simulation (corresponding to Figure 5). Here, the number of principal components are $\kappa = 5,10,20,30,40$ corresponding to the five cases $J = 1,2,4,6,8$, respectively, to ensure 95% of variance explained. Here, eigenvalues of the concatenated high-frequency Z decay at an exponential order $\lambda_k = O(e^{-\zeta k})$ with $\zeta = 0.6$.

frequency variable Z decays at an exponential order. We find that the first five principal components explain over 95% of variance in all the five cases, as shown in Figure 3. Therefore, we set the number of principal components $\kappa = 5$ and plot the results in Figure 4.

In the second setting, the J high-frequency variables are independent with each other. For each j, all the elements in Z_j are dependent and eigenvalues of Z_j decay at the same exponential order as the eigenvalues of the concatenated high-frequency variable Z in the first setting. Therefore, more principal components are needed to guarantee that the number of variance explained exceeds 95%, as J increases. Specifically, when $J = 1,2,4,6,8$ high-frequency variables, the corresponding κ is given by 5, 10, 20, 30, 40 accordingly. See Figure 3 for details. The expected returns of all estimated optimal policies are plotted in Figure 5.

From Figures 4 and 5, it can be seen that the proposed policy π_K^{PCA} always achieves a larger value than the three baseline policies. It is true even in the two scenarios with very limited data size, which proves the robustness of our proposed method. Meanwhile, π_K^{ALL} and π_K^{BOTTLE} perform comparably. The value of both of them are significantly affected by the training size n. In addition, π_K^{AVE} outperforms π_K^{ALL} and π_K^{BOTTLE} in small samples but performs worse than the two policies when the sample size is large. In the second setting, π_K^{ALL} and π_K^{BOTTLE} tend to perform much better than π_K^{AVE} when $J = 4,6,8$, since averaging over several high-frequency variables will lose more relevant information for policy learning.

Finally, we conduct an additional simulation study with large training datasets where $N = 200$ or 4000 and $T = 120$. This setting might be unrealistic in an mHealth dataset. It is included only to test the performance of π_K^{ALL}. As π_K^{ALL} is consistent as well, we anticipate that the difference between the value under π_K^{ALL} and the proposed policy will be negligible as the sample size grows to infinity. Figure 6 depicts the results. As expected, we observe no significant difference between π^{PCA} and π^{ALL} when $N \geq 200$.

The time complexity of obtaining the principal component scores for all high dimensional vectors is $O(m^3 + nm^2)$. Denote the maximum width and depth of hidden layers in the neural network of $\tilde{Q}$ as d and L, respectively. Assuming that the maximum number of iterations of fitted Q-iteration is K and that the number of training epochs for the neural network in each iteration is M, the time complexity of obtaining $\tilde{Q}_K$ is given by $O(nKM|\mathcal{A}|[(m_0 + \kappa)d + Ld^2])$ (see Section 6.5.7 of Goodfellow et al., 2016). Therefore, the total time complexity of Algorithm 1 is

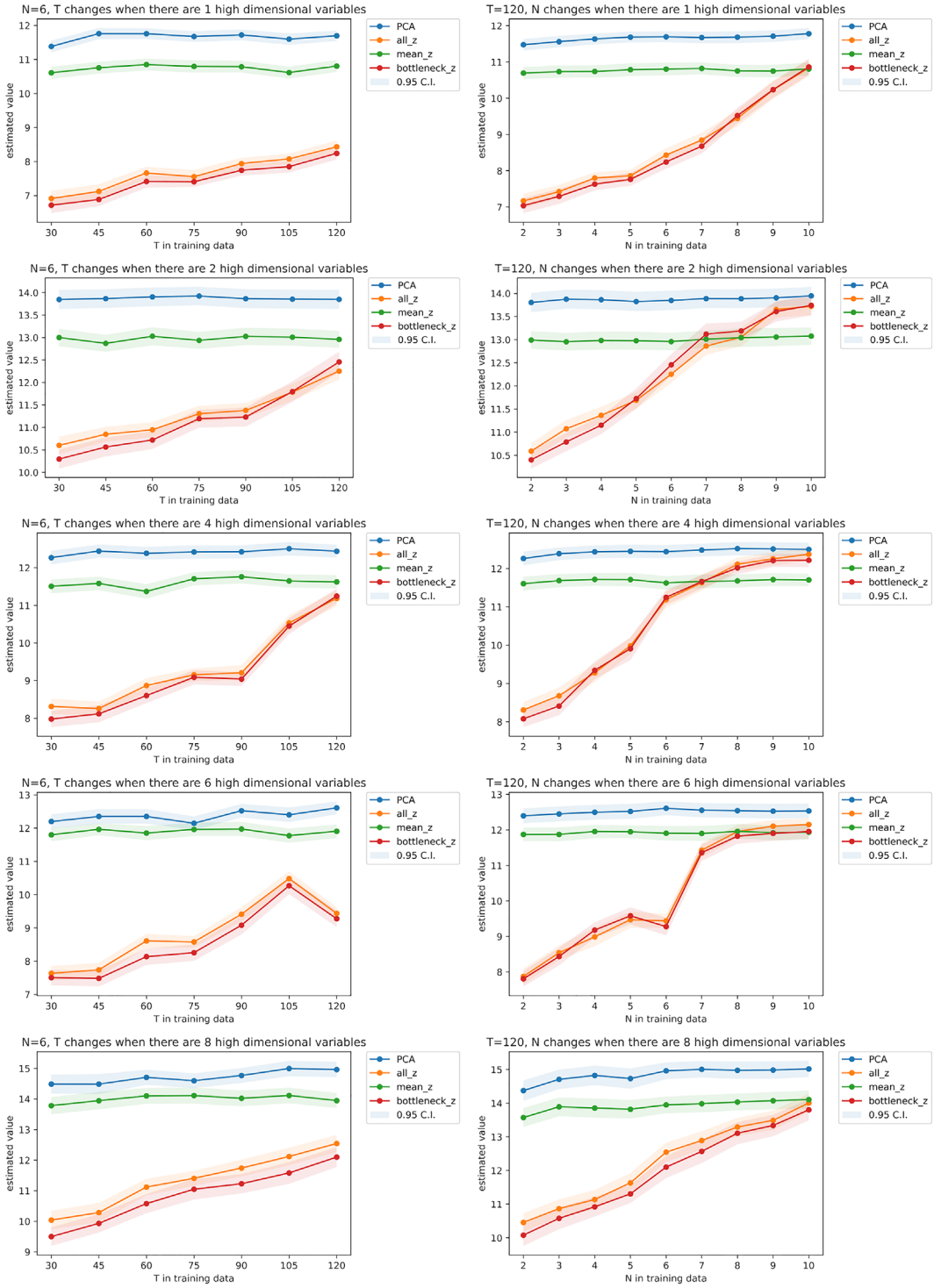

FIGURE 4 **First setting** in simulation. Left: training data with $N = 6$ and $T = 30,45,60,75,90,105,120$; right: training data with $T = 120$ and $N = 2,3,4,5,6,7,8,9,10$ when there are 1, 2, 4, 6, 8 variables with dimension 27 (shaded area is 95% confidence interval). In the legend, "PCA" refers to π_K^{PCA}; "all z" refers to π_K^{ALL}; "bottleneck z" refers to π_K^{BOTTLE}; "mean z" refers to π_K^{AVE}. The leftmost dots in the plots (corresponding to $T =$ 30 or $N = 2$) represent the cases with limited observations.

$O(nKM|\mathcal{A}|[(m_0+\kappa)d+Ld^2]+m^3+nm^2)$. In the simulation study, the training time of neural network is the bottleneck of the total computation time, which dominates the time required for calculating the principal components. In the experiments with $m = 27$ and $\kappa = 5$ (corresponding to the settings plotted in the first rows of Figures 4 and 5), the average running time (on a single CPU) of Algorithm 1 for an offline data with $N = 2, T =$ 120 and $N = 10, T = 120$ is 122 and 834 s, respectively.

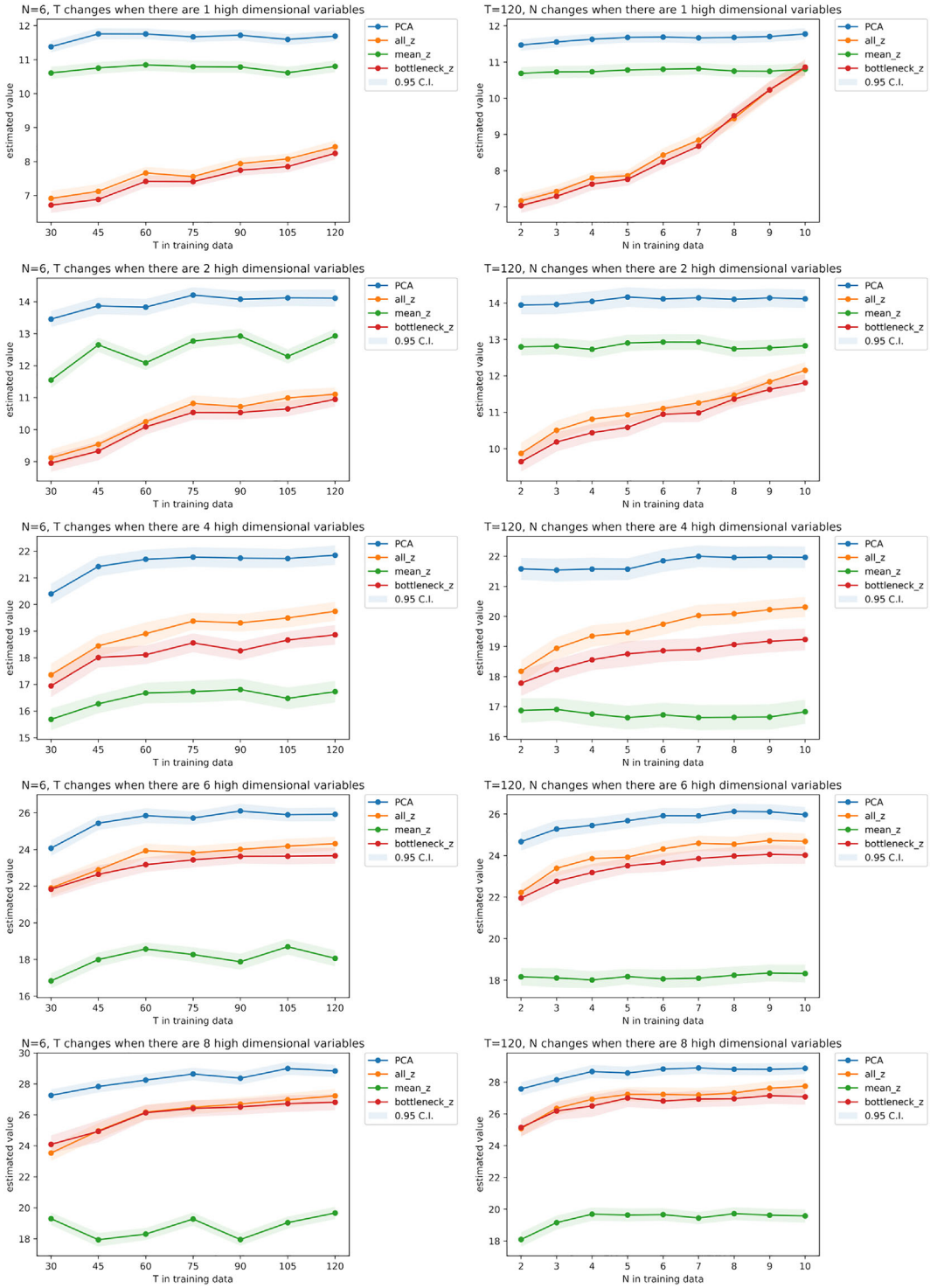

FIGURE 5 **Second setting** in simulation. Left: training data with $N = 6$ and $T = 30,45,60,75,90,105,120$; right: training data with $T = 120$ and $N = 2,3,4,5,6,7,8,9,10$ when there are 1, 2, 4, 6, 8 variables with dimension 27 (shaded area is 95% confidence interval). In the legend, "PCA" refers to π_K^{PCA}; "all z" refers to π_K^{ALL}; "bottleneck z" refers to π_K^{BOTTLE}; "mean z" refers to π_K^{AVE}. The leftmost dots in the plots (corresponding to $T =$ 30 or $N = 2$) represent the cases with limited observations.

5.3 | Application on OhioT1DM Dataset

We apply the proposed Algorithm 1 on the updated OhioT1DM Dataset by Marling and Bunescu (2020). In the real data case, we would still like to compare the behaviors of the four policies: $\pi_K^{PCA}, \pi_K^{ALL}, \pi_K^{BOTTLE}$, and π_K^{AVE}. OhioT1DM Dataset contains medical information of 12 patients

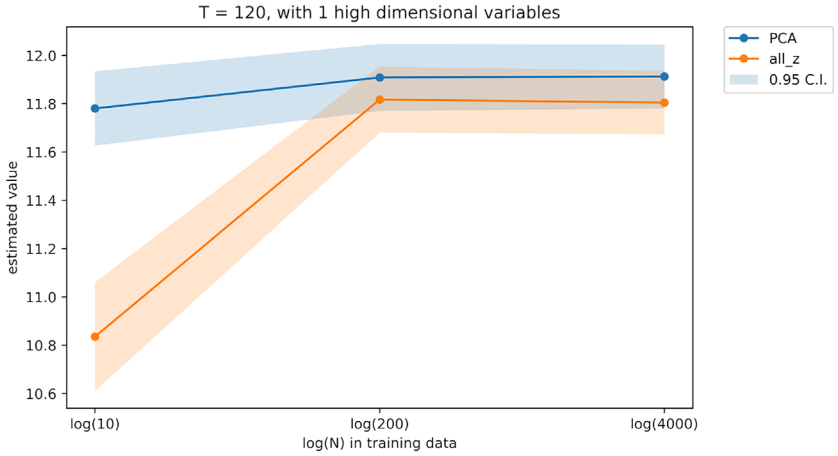

FIGURE 6 Performance of π^{PCA} and π^{ALL} as training data size n goes to extremely large (T fixed to be 120, N from 10 to 200 and 4000). The x-axis presents log$N$. In this figure, shaded area is 95% confidence interval.In the legend, "PCA" refers to π_K^{PCA}; "all z" refers to π_K^{ALL}.

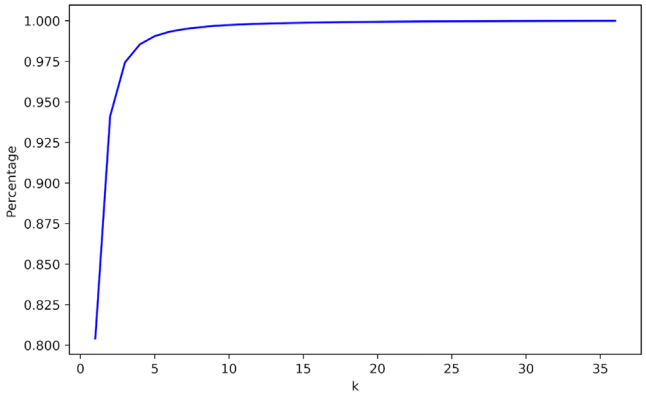

FIGURE 7 Proportion of variance explained by the first κ principal components for CGM blood glucose level in OhioT1DM data.

suffering from type-I diabetes, including the CGM blood glucose levels of the patients, insulin doses applied during this period, self-reported information of meals and exercises, and other variables recorded by mobile phone apps and physiological sensors. The high-frequency variables in the OhioT1DM Dataset, such as CGM blood glucose levels, are recorded every 5 min. The data for exercises and meals are collected with a much lower frequency, say recorded every few hours. Moreover, considering the basal insulin rate of in this dataset, although this variable is also collected every 5 min, it usually remains a constant for several hours in a day. Thus, the basal insulin rate can also be regarded as a low-frequency scalar variable by taking the average of it. The time period between two decision points is set as 3 h, as we only consider non-emergency situations where patients don't need to take bolus injection promptly. In other studies using the OhioT1DM Dataset, the treatment decision frequency is also set to be much lower than the recording frequency of CGM blood glucose levels (see, e.g., Shi et al., 2022; Zhou et al., 2021; Zhu et al., 2020).

For the low-frequency covariate $X_{i,t} = (X_{i,t}^{(1)}, X_{i,t}^{(2)}), X_{i,t}^{(1)}$ is constructed based on the ith patient's self-reported carbohydrate estimate for the meal during the past 3-h interval $[t-1, t)$. The second scalar variable in $X_{i,t}$ is defined as the average of the basal rate of insulin dose during the past interval $[t-1, t)$. We consider one high-frequency element, $Z_{i,t}$, which contains CGM blood glucose levels recorded every 5 min during the past 3 h (its dimension is $m = 36$). The action variable $A_{i,t}$ is set to 1 when the total amount of insulin delivered to the i-th patient is greater than one unit in the past interval and 0 otherwise. The response variable $R_{i,t}$ is defined according to the Index of Glycemic Control (IGC Rodbard, 2009), which is a nonlinear function of the blood glucose levels in the following stage. A higher IGC value indicates that the blood glucose level of this patient stays close or falls in to the proper range of glucose level.

In this study, $\kappa = 5$ is selected when training π_K^{PCA}, as the proportion of variance explained by the first 5 principal components is over 99%, as is shown in Figure 7. The ReLU network here is with two hidden layers and width $d_i = 6, i = 1,2$ (dropout layers with 10% dropout rate added

TABLE 1 Estimate of value difference of four policies.

Difference	$\pi_k^{PCA} - \pi_k^{ALL}$	$\pi_k^{PCA} - \pi_k^{AVE}$	$\pi_k^{PCA} - \pi_k^{BOTTLE}$
Mean	1.163	1.399	1.153
Margin of error	0.181	0.224	0.145

TABLE 2 Value estimate of the four policies.

Value Estimate	π_k^{PCA}	π_k^{ALL}	π_k^{AVE}	π_k^{BOTTLE}
Mean	-8.762	-9.926	-10.162	-9.916
Standard error	0.055	0.096	0.121	0.086

between layer 1, layer 2, and the output layer). To estimate the value $V^{\pi}(x,z)$ of the four policies, we use the fitted Q evaluation algorithm proposed by Le et al. (2019). When applying the fitted Q evaluation algorithm, a random forest model is used to fit the estimated Q-function of the policy to be evaluated. By dividing 12 patients into a training set of nine patients and testing set of three patients, there are 220 repetitions with different patient combinations. In each repetition, the data of nine patients is used to train the policy and fit the random forest for fitted Q evaluation corresponding to this policy. The data of the other three patient is used for approximating the value of the policy using the estimated Q-function from fitted Q evaluation. The sample mean of estimated values from all 220 repetitions is taken as our main result and the standard errors are used to construct the margin of error. To compare the performance of our proposed policies π_K^{PCA} against π_K^{ALL}, π_K^{AVE}, and π_K^{BOTTLE}, we present the difference of estimated values of π^{PCA} and the three other policies in Table 1, where margin of error is standard error of the mean difference multiplied by the critical value 1.96. The estimated values of the four policies is in Table 2.

Based on the result, it can be shown that the estimated value of π_K^{PCA} is higher than all three baselines. The policy π_K^{AVE} obtained by using the average of CGM blood glucose levels is commonly used in literature (Zhou et al., 2021; Zhu et al., 2020). The less plausible performance of π_K^{AVE} is probably due to the information loss by simply replacing the CGM blood glucose levels with its average. On the other hand, the size of training data is relatively small, as we didn't use all the data recorded in eight weeks due to the large chunks of missing values. Eventually training data from about five consecutive weeks are used for training DTR policies. In such scenarios, using the original high-frequency vector $Z_{i,t}$ will significantly increase the complexity of the ReLU network structure, such that the number of parameters to be trained is too large compared to the size of training data. Thus, π_K^{ALL} and π_K^{BOTTLE} cannot outperform π_K^{PCA} where input dimension is reduced by PCA. The results shown in Tables 1 and 2 agree with the results in Section 5.2.

6 | DISCUSSIONS

In summary, we propose a deep spectral fitted Q-iteration algorithm to handle mixed frequency data in infinite horizon settings. The algorithm relies on the use of PCA for dimension reduction and the use of deep neural networks to capture the nonlinearity in the high-dimensional system. In theory, we establish a regret bound for the estimated optimal policy. Our theorem provides an asymptotic guideline for selecting the number of principal components. In empirical studies, we demonstrate the superiority of the proposed algorithm over baseline methods without dimension reduction or use ad hoc summaries of the state. We further offer practical guidelines to select the number of principal components.

The current focus of our work is on training the policy with deep Q-learning offline on a finite treatment set. In the future, we plan to extend our proposed method to an online version. First, notice that the covariance matrix of Z can be updated in an online manner (see, e.g., Dasgupta & Hsu, 2007). Second, the fitted Q-iteration algorithm is originally developed in the online setting (Riedmiller, 2005). Specifically, at each iteration, we adopt an ϵ-greedy policy to adaptively generate the data, calculate the principle components based on the estimated covariance matrix, and fit a deep neural network model for the Q-function using these data. We repeat this procedure until convergence. It is worthwhile to investigate the asymptotic properties of the resulting algorithm in future research.

Another potential research direction is to extended the current approach to handle a continuous action space. In this case, the fitted Q-iteration algorithm needs to be replaced by some policy-based learning methods (e.g., actor-critic). Nonetheless, the principal component analysis is equally applicable to obtain a parsimonious representation of the state as the input in both the actor and the critic models. It would be interesting to study the theoretical properties of the resulting algorithm in future work. Furthermore, it is worthwhile to explore the effectiveness of other dimension reduction methods, such as supervised PCA (Bair et al., 2006), Isomap (Tenenbaum et al., 2000), and t-SNE (Van der Maaten & Hinton, 2008) in future studies.

DATA AVAILABILITY STATEMENT

Python code for reproducing the simulation results is included in the Supporting Information.

ORCID

Yuhe Gao https://orcid.org/0000-0001-5359-0604

REFERENCES

Arulkumaran, K., Deisenroth, M. P., Brundage, M., & Bharath, A. A. (2017). Deep reinforcement learning: A brief survey. *IEEE Signal Processing Magazine*, *34*(6), 26–38.

Audibert, J. Y., & Tsybakov, A. B. (2007). Fast learning rates for plug-in classifiers. *The Annals of Statistics*, *35*(2), 608–633.

Bair, E., Hastie, T., Paul, D., & Tibshirani, R. (2006). Prediction by supervised principal components. *Journal of the American Statistical Association*, *101*(473), 119–137.

Chen, E. Y., Song, R., & Jordan, M. I. (2022). Reinforcement learning with heterogeneous data: Estimation and inference. arXiv preprint arXiv:2202.00088.

Chen, J., & Jiang, N. (2019). Information-theoretic considerations in batch reinforcement learning. *International Conference on Machine Learning*, *2019*, 1042–1051.

Chen, L., Lu, K., Rajeswaran, A., Lee, K., Grover, A., Laskin, M., Abbeel, P., Srinivas, A., & Mordatch, I. (2021). Decision transformer: Reinforcement learning via sequence modeling. *Advances in Neural Information Processing Systems*, *34*, 15084–15097.

Ciarleglio, A., Petkova, E., Ogden, R. T., & Tarpey, T. (2015). Treatment decisions based on scalar and functional baseline covariates. *Biometrics*, *71*(4), 884–894.

Ciarleglio, A., Petkova, E., Ogden, T., & Tarpey, T. (2018). Constructing treatment decision rules based on scalar and functional predictors when moderators of treatment effect are unknown. *Journal of the Royal Statistical Society. Series C, Applied statistics*, *67*(5), 1331.

Ciarleglio, A., Petkova, E., Tarpey, T., & Ogden, R. T. (2016). Flexible functional regression methods for estimating individualized treatment rules. *Statistical (International Statistical Institute)*, *5*(1), 185–199.

Crambes, C., & Mas, A. (2013). Asymptotics of prediction in functional linear regression with functional outputs. *Bernoulli*, *2013*, 2627–2651.

Curran, W., Brys, T., Aha, D., Taylor, M., & Smart, W. D. (2016). Dimensionality reduced reinforcement learning for assistive robots. In *2016 AAAI Fall Symposium Series*.

Curran, W., Brys, T., Taylor, M., & Smart, W. (2015). Using PCA to efficiently represent state spaces. arXiv preprint arXiv:1505.00322.

Dabney, W., Rowland, M., Bellemare, M. G., & Munos, R. (2018). Distributional reinforcement learning with quantile regression. Thirty-Second AAAI Conference on Artificial Intelligence.

Dasgupta, S., & Hsu, D. (2007). On-line estimation with the multivariate gaussian distribution. In *Learning theory: 20th Annual Conference on Learning Theory, COLT 2007, San Diego, CA, USA; June 13-15, 2007. Proceedings 20*, Springer, pp. 278–292.

Dickey, D. A., & Fuller, W. A. (1979). Distribution of the estimators for autoregressive time series with a unit root. *Journal of the American Statistical Association*, *74*(366a), 427–431.

Ernst, D., Geurts, P., & Wehenkel, L. (2005). Tree-based batch mode reinforcement learning. *Journal of Machine Learning Research*, *6*, 503–556.

Ertefaie, A., McKay, J. R., Oslin, D., & Strawderman, R. L. (2021). Robust q-learning. *Journal of the American Statistical Association*, *116*(533), 368–381.

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

Fan, J., Wang, Z., Xie, Y., & Yang, Z. (2020). A theoretical analysis of deep q-learning. *Learning for Dynamics and Control*, *2020*, 486–489.

Fang, E. X., Wang, Z., & Wang, L. (2022). Fairness-oriented learning for optimal individualized treatment rules. *Journal of the American Statistical Association*, *2022*, 1–14.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*: MIT Press. http://www.deeplearningbook.org

Guan, Q., Reich, B. J., Laber, E. B., & Bandyopadhyay, D. (2020). Bayesian nonparametric policy search with application to periodontal recall intervals. *Journal of the American Statistical Association*, *115*(531), 1066–1078.

Guestrin, C., Patrascu, R., & Schuurmans, D. (2002). Algorithm-directed exploration for model-based reinforcement learning in factored MDPs. In *ICML*, Citeseer, pp. 235–242.

Jameson, J. L., & Longo, D. L. (2015). Precision medicine personalized, problematic, and promising. *Obstetrical & Gynecological Survey*, *70*(10), 612–614.

Janner, M., Fu, J., Zhang, M., & Levine, S. (2019). When to trust your model: Model-based policy optimization. *Advances in Neural Information Processing Systems*, *32*.

Janner, M., Li, Q., & Levine, S. (2021). Offline reinforcement learning as one big sequence modeling problem. *Advances in Neural Information Processing Systems*, *34*, 1273–1286.

Jirak, M. (2016). Optimal Eigen expansions and uniform bounds. *Probability Theory and Related Fields*, *166*(3), 753–799.

Kallus, N., & Uehara, M. (2022). Efficiently breaking the curse of horizon in off-policy evaluation with double reinforcement learning. Operations Research.

Kosorok, M. R., & Laber, E. B. (2019). Precision medicine. *Annual Review of Statistics and Its Application*, *6*, 263–286.

Koutnk, J., Cuccu, G., Schmidhuber, J., & Gomez, F. (2013). Evolving large-scale neural networks for vision-based reinforcement learning. In *Proceedings of the 15th Annual Conference on Genetic and Evolutionary Computation*, pp. 1061–1068.

Kwiatkowski, D., Phillips, P. eterC. B., Schmidt, P., & Shin, Y. (1992). Testing the null hypothesis of stationarity against the alternative of a unit root: How sure are we that economic time series have a unit root?. *Journal of Econometrics*, *54*(1-3), 159–178.

Laber, E. B., & Staicu, A. M. (2018). Functional feature construction for individualized treatment regimes. *Journal of the American Statistical Association*, *113*(523), 1219–1227.

Lai, H., Shen, J., Zhang, W., & Yu, Y. (2020). Bidirectional model-based policy optimization. In *International conference on machine learning*, PMLR, pp. 5618–5627.

Le, H., Voloshin, C., & Yue, Y. (2019). Batch policy learning under constraints. In *International Conference on Machine Learning*, pp. 3703–3712.

Li, G., Wei, Y., Chi, Y., Gu, Y., & Chen, Y. (2020). Breaking the sample size barrier in model-based reinforcement learning with a generative model. *Advances in Neural Information Processing Systems*, *33*, 12861–12872.

Li, Y., Wang, N., & Carroll, R. J. (2013). Selecting the number of principal components in functional data. *Journal of the American Statistical Association*, *108*(504), 1284–1294.
Li, Y., Wang, C., Cheng, G., & Sun, W. W. (2022). Rate-optimal contextual online matching bandit. arXiv preprint arXiv:2205.03699.
Liao, P., Klasnja, P., & Murphy, S. (2021). Off-policy estimation of long-term average outcomes with applications to mobile health. *Journal of the American Statistical Association*, *116*(533), 382–391.
Liao, P., Qi, Z., Klasnja, P., & Murphy, S. (2020). Batch policy learning in average reward Markov decision processes. arXiv preprint arXiv:2007.11771.
Luckett, D. J., Laber, E. B., Kahkoska, A. R., Maahs, D. M., Mayer-Davis, E., & Kosorok, M. R. (2020). Estimating dynamic treatment regimes in mobile health using v-learning. *Journal of the American Statistical Association*, *115*(530), 692–706. https://doi.org/10.1080/01621459.2018.1537919
Luedtke, A. R., & van der Laan, M. J. (2016). Statistical inference for the mean outcome under a possibly non-unique optimal treatment strategy. *The Annals of Statistics*, *44*(2), 713–742. https://doi.org/10.1214/15-AOS1384
Luo, F.-M., Xu, T., Lai, H., Chen, X.-H., Zhang, W., & Yu, Y. (2022). A survey on model-based reinforcement learning. arXiv preprint arXiv:2206.09328.
Marling, C., & Bunescu, R. (2020). The OhioT1DM dataset for blood glucose level prediction: Update 2020. *CEUR Workshop Proceedings*, *2675*, 71–74.
McKeague, I. W., & Qian, M. (2014). Estimation of treatment policies based on functional predictors. *Statistica Sinica*, *24*(3), 1461.
Mnih, V., Badia, A. P., Mirza, M., Graves, A., Lillicrap, T., Harley, T., Silver, D., & Kavukcuoglu, K. (2016). Asynchronous methods for deep reinforcement learning. In *International Conference on Machine Learning*, pp. 1928–1937.
Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, R., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., & Petersen, S. (2015). Human-level control through deep reinforcement learning. *Nature*, *518*(7540), 529–533.
Mo, W., Qi, Z., & Liu, Y. (2021). Learning optimal distributionally robust individualized treatment rules. *Journal of the American Statistical Association*, *116*(534), 659–674.
Murao, H., & Kitamura, S. (1997). Q-learning with adaptive state segmentation (QLASS). In *In Proceedings 1997 IEEE International Symposium on Computational Intelligence in Robotics and Automation CIRA'97.'Towards New Computational Principles for Robotics and Automation'*, pp. 179–184.
Murphy, S. A. (2003). Optimal dynamic treatment regimes. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, *65*(2), 331–355.
Nie, X., Brunskill, E., & Wager, S. (2021). Learning when-to-treat policies. *Journal of the American Statistical Association*, *116*(533), 392–409.
Parisi, S., Ramstedt, S., & Peters, J. (2017). Goal-driven dimensionality reduction for reinforcement learning. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, pp. 4634–4639.
Puterman, M. L. (2014). *Markov decision processes: discrete stochastic dynamic programming*: John Wiley & Sons.
Qi, Z., & Liu, Y. (2018). D-learning to estimate optimal individual treatment rules. *Electronic Journal of Statistics*, *12*(2), 3601–3638.
Qian, M., & Murphy, S. A. (2011). Performance guarantees for individualized treatment rules. *Annals of Statistics*, *39*(2), 1180.
Radford, A., Narasimhan, K., Salimans, T., & Sutskever, I. (2018). Improving language understanding by generative pre-training.
Ramprasad, P., Li, Y., Yang, Z., Wang, Z., Sun, W. W., & Cheng, G. (2022). Online bootstrap inference for policy evaluation in reinforcement learning. *Journal of the American Statistical Association*, *2022*, 1–14.
Reiß, M., & Wahl, M. (2020). Nonasymptotic upper bounds for the reconstruction error of PCA. *The Annals of Statistics*, *48*(2), 1098–1123.
Riedmiller, M. (2005). Neural fitted Q iteration first experiences with a data efficient neural reinforcement learning method. *European Conference on Machine Learning*, *2005*, 317–328.
Rodbard, D. (2009). Interpretation of continuous glucose monitoring data: Glycemic variability and quality of glycemic control. *Diabetes Technology & Therapeutics*, *11*(S1), S–55.
Said, S. E., & Dickey, D. A. (1984). Testing for unit roots in autoregressive-moving average models of unknown order. *Biometrika*, *71*(3), 599–607.
Schmidt-Hieber, J. (2020). Nonparametric regression using deep neural networks with ReLU activation function. *The Annals of Statistics*, *48*(4), 1875–1897.
Schulman, J., Levine, S., Abbeel, P., Jordan, M., & Moritz, P. (2015). Trust region policy optimization. *International Conference on Machine Learning*, *2015*, 1889–1897.
Schulman, J., Moritz, P., Levine, S., Jordan, M., & Abbeel, P. (2015). High-dimensional continuous control using generalized advantage estimation. arXiv preprint arXiv:1506.02438.
Shi, C., Song, R., & Lu, W. (2021). Concordance and value information criteria for optimal treatment decision. *The Annals of Statistics*, *49*(1), 49–75.
Shi, C., Song, R., Lu, W., & Fu, B. (2018). Maximin projection learning for optimal treatment decision with heterogeneous individualized treatment effects. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, *80*(4), 681–702.
Shi, C., Wan, R., Chernozhukov, V., & Song, R. (2021). Deeply-debiased off-policy interval estimation. In *International conference on machine learning*, PMLR, pp. 9580–9591.
Shi, C., Wan, R., Song, R., Lu, W., & Leng, L. (2020). Does the Markov decision process fit the data: Testing for the Markov property in sequential decision making. In *International Conference on Machine Learning*, PMLR, pp. 8807–8817.
Shi, C., Wang, X., Luo, S., Zhu, H., Ye, J., & Song, R. (2021). Dynamic causal effects evaluation in a/b testing with a reinforcement learning framework. Journal of the American Statistical Association, accepted.
Shi, C., Zhang, S., Lu, W., & Song, R. (2022). Statistical inference of the value function for reinforcement learning in infinite horizon settings. *Journal of the Royal Statistical Society: Series B*, *84*, 765–793.
Song, R., Wang, W., Zeng, D., & Kosorok, M. R. (2015). Penalized q-learning for dynamic treatment regimens. *Statistica Sinica*, *25*(3), 901.
Sprague, N. (2007). Basis iteration for reward based dimensionality reduction. In *2007 IEEE 6th International Conference on Development and Learning*, pp. 187–192.
Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, *15*(1), 1929–1958.
Staicu, A.-M., Li, Y., Crainiceanu, C. M., & Ruppert, D. (2014). Likelihood ratio tests for dependent data with applications to longitudinal and functional data analysis. *Scandinavian Journal of Statistics*, *41*(4), 932–949.
Sutton, R. S., & Barto, A. G. (2018). *Reinforcement learning: An introduction*: MIT Press.
Tenenbaum, J. B., Silva, V., & Langford, J. C. (2000). A global geometric framework for nonlinear dimensionality reduction. *Science*, *290*(5500), 2319–2323.
Tsiatis, A. A., Davidian, M., Holloway, S. T., & Laber, E. B. (2019). *Dynamic treatment regimes: Statistical methods for precision medicine*: Chapman and Hall/CRC.
Tsybakov, A. B. (2004). Optimal aggregation of classifiers in statistical learning. *The Annals of Statistics*, *32*(1), 135–166.

Uehara, M., Imaizumi, M., Jiang, N., Kallus, N., Sun, W., & Xie, T. (2021). Finite sample analysis of minimax offline reinforcement learning: Completeness, fast rates and first-order efficiency. arXiv preprint arXiv:2102.02981.

Uehara, M., Kiyohara, H., Bennett, A., Chernozhukov, V., Jiang, N., Kallus, N., Shi, C., & Sun, W. (2022). Future-dependent value-based off-policy evaluation in POMDPs. arXiv preprint arXiv:2207.13081.

Van der Maaten, L., & Hinton, G. (2008). Visualizing data using t-SNE. *Journal of machine learning research*, *9*, 11.

Van Hasselt, H., Guez, A., & Silver, D. (2016). Deep reinforcement learning with double q-learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, *30*.

Villaflor, A. R., Huang, Z., Pande, S., Dolan, J. M., & Schneider, J. (2022). Addressing optimism bias in sequence modeling for reinforcement learning. In *International Conference on Machine Learning*, PMLR, pp. 22270–22283.

Wang, L., Laber, E. B., & Witkiewitz, K. (2017). Sufficient Markov decision processes with alternating deep neural networks. arXiv preprint arXiv: 1704.07531.

Wang, Z., Bapst, V., Heess, N., Mnih, V., Munos, R., Kavukcuoglu, K., & de Freitas, N. (2017). Sample efficient actor-critic with experience replay. ICLR.

Whiteson, S., Taylor, M. E., & Stone, P. (2007). Adaptive tile coding for value function approximation.

Xu, Z., Laber, E., Staicu, A.-M., & Severus, E. (2020). Latent-state models for precision medicine. arXiv preprint arXiv:2005.13001.

Yao, F., Müller, H.-G., & Wang, J.-L. (2005). Functional data analysis for sparse longitudinal data. *Journal of the American statistical Association*, *100*(470), 577–590.

Zhang, B., Tsiatis, A. A., Laber, E. B., & Davidian, M. (2013). Robust estimation of optimal dynamic treatment regimes for sequential treatment decisions. *Biometrika*, *100*(3), 681–694.

Zhang, X., & Wang, J. L. (2016). From sparse to dense functional data and beyond. *The Annals of Statistics*, *44*(5), 2281–2321.

Zhang, Y., Laber, E. B., Davidian, M., & Tsiatis, A. A. (2018). Interpretable dynamic treatment regimes. *Journal of the American Statistical Association*, *113*(524), 1541–1549.

Zhao, Y.-Q., Zeng, D., Laber, E. B., & Kosorok, M. R. (2015). New statistical learning methods for estimating optimal dynamic treatment regimes. *Journal of the American Statistical Association*, *110*(510), 583–598.

Zhou, W., Zhu, R., & Qu, A. (2021). Estimating optimal infinite horizon dynamic treatment regimes via pt-learning. arXiv preprint arXiv:2110.10719.

Zhou, W., Zhu, R., & Qu, A. (2022). Estimating optimal infinite horizon dynamic treatment regimes via pT-learning. Journal of the American Statistical Association, accepted.

Zhu, L., Lu, W., & Song, R. (2020). Causal effect estimation and optimal dose suggestions in mobile health. In *International conference on machine learning*, PMLR, pp. 11588–11598.

SUPPORTING INFORMATION

Additional supporting information can be found online in the Supporting Information section at the end of this article.

How to cite this article: Gao, Y., Shi, C., & Song, R. (2023). Deep spectral Q-learning with application to mobile health. *Stat*, *12*(1), e564. https://doi.org/10.1002/sta4.564