Learning from Conditional Distributions via Dual Embeddings

Bo Dai
Georgia Tech

Niao He
UIUC

Yunpeng Pan
Georgia Tech

Byron Boots
Georgia Tech

Le Song
Georgia Tech

Abstract

Many machine learning tasks, such as learning with invariance and policy evaluation in reinforcement learning, can be characterized as problems of *learning from conditional distributions*. In such problems, each sample x itself is associated with a conditional distribution $p(z|x)$ represented by samples $\{z_i\}_{i=1}^M$, and the goal is to learn a function f that links these conditional distributions to target values y. These problems become very challenging when we only have limited samples or in the extreme case only one sample from each conditional distribution. Commonly used approaches either assume that z is independent of x, or require an overwhelmingly large set of samples from each conditional distribution.

To address these challenges, we propose a novel approach which employs a new *min-max reformulation* of the learning from conditional distribution problem. With such new reformulation, we only need to deal with the *joint distribution* $p(z, x)$. We also design an efficient learning algorithm, *Embedding-SGD*, and establish theoretical sample complexity for such problems. Finally, our numerical experiments, on both synthetic and real-world datasets, show that the proposed approach can significantly improve over existing algorithms.

1 Introduction

We address the problem of *learning from conditional distributions* where the goal is to learn a function that links conditional distributions to target variables. Specifically, we are provided input samples $\{x_i\}_{i=1}^N \in \mathcal{X}^N$ and their corresponding responses $\{y_i\}_{i=1}^N \in \mathcal{Y}^N$. For each $x \in \mathcal{X}$, there is an associated conditional distribution $p(z|x) : \mathcal{Z} \times \mathcal{X} \to \mathbb{R}$. However, we cannot access entire conditional distributions $\{p(z|x_i)\}_{i=1}^N$ directly; rather, we only observe a limited number of samples or, in the extreme case, only *one sample* from each conditional distribution $p(z|x)$. The task is to learn a function f which links the conditional distribution $p(z|x)$ to target $y \in \mathcal{Y}$ by minimizing the expected loss:

$$\min_{f \in \mathcal{F}} \; L(f) = \mathbb{E}_{x,y} \left[\ell \left(y, \mathbb{E}_{z|x} \left[f(z, x) \right] \right) \right] \tag{1}$$

where $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ is a convex loss function. The function space $\mathcal{F}$ can be very general, but we focus on the case when $\mathcal{F}$ is a reproducing kernel Hilbert space (RKHS) in the main text, namely, $\mathcal{F} = \{f : \mathcal{Z} \times \mathcal{X} \to \mathbb{R} \,|\, f(z, x) = \langle f, \psi(z, x) \rangle\}$ where $\psi(z, x)$ is a suitably chosen (nonlinear) feature map. Please refer to Appendix E for the extension to arbitrary function approximators, *e.g.*, random features and neural networks.

The problem of learning from conditional distributions appears in many different tasks. For example:

- **Learning with invariance.** Incorporating priors on invariance into the learning procedure is crucial for computer vision (Niyogi et al., 1998), speech recognition (Anselmi et al., 2013) and many other applications. The goal of invariance learning is to estimate a function which minimizes the expected risk while at the same time preserving consistency over a group of operations $g = \{g_j\}_{j=1}^\infty$. Mroueh et al. (2015) shows that this can be accomplished by solving the following optimization problem
$$\min_{f \in \tilde{\mathcal{H}}} \mathbb{E}_{x,y}[\ell(y, \mathbb{E}_{z|x \sim \mu(g(x))}[\langle f, \psi(z) \rangle_{\tilde{\mathcal{H}}}])] + (\nu/2)\|f\|_{\tilde{\mathcal{H}}}^2 \tag{2}$$
where $\tilde{\mathcal{H}}$ is the RKHS corresponding to kernel $\tilde{k}$ with the feature map $\psi(\cdot)$, $\nu > 0$ is the regularization parameter. Obviously, the above optimization (2) is a special case of (1). In this case, z stands for possible variation of data x through conditional probability given by some normalized Haar measure $\mu(g(x))$. Due to computation and memory constraints, one can only afford to generate a few virtual samples from each data point x.

- **Policy evaluation in reinforcement learning.** Policy evaluation is a fundamental task in reinforcement learning. Given a policy $\pi(a|s)$, which is a distribution over

Proceedings of the 20^{th} International Conference on Artificial Intelligence and Statistics (AISTATS) 2017, Fort Lauderdale, Florida, USA. JMLR: W&CP volume 54. Copyright 2017 by the author(s).

action space condition on current state s, the goal is to estimate the value function $V^\pi(\cdot)$ over the state space. $V^\pi(s)$ is the fixed point of the Bellman equation

$$V^\pi(s) = \mathbb{E}_{s'|a,s}[R(s,a) + \gamma V^\pi(s')],$$

where $R(s,a) : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is a reward function and $\gamma \in (0,1)$ is the discount factor. Therefore, the value function can be estimated from data by minimizing the mean-square Bellman error (Baird, 1995; Sutton et al., 2008):

$$\min_{V^\pi} \mathbb{E}_{s,a}\left[\left(R(s,a) - \mathbb{E}_{s'|a,s}\left[V^\pi(s) - \gamma V^\pi(s')\right]\right)^2\right]. \quad (3)$$

Restricting the policy to lie in some RKHS, this optimization is clearly a special case of (1) by viewing $((s,a), R(s,a), s')$ as (x,y,z) in (1). Here, given state s and the the action $a \sim \pi(a|s)$, the successor state s' comes from the transition probability $p(s'|a,s)$. Due to the online nature of MDPs, we usually observe only one successor state s' for each action a given s, *i.e.*, only one sample from the conditional distribution given s, a.

Challenges. Despite many learning problems in the form of (1), solving such problems remains very challenging for two reasons: (**i**), we often have limited samples or, in the extreme case, only one sample from each conditional distribution $p(z|x)$, making it difficult to accurately estimate the conditional expectation. (**ii**), the conditional expectation is nested inside the loss function, making the problem quite different from the traditional stochastic optimization setting. This type of problem is called *compositional stochastic programming*, and very few results have been established in this domain.

Related work. A simple option to address (1) is using sample average approximation (SAA), and thus, instead solving

$$\min_{f\in\mathcal{F}} \frac{1}{N}\sum_{i=1}^N\left[\ell\left(y_i, \frac{1}{M}\sum_{j=1}^M f(z_{ij}, x_i)\right)\right],$$

where $\{(x_i,y_i)\}_{i=1}^N \sim p(x,y)$, and $\{z_{ij}\}_{j=1}^M \sim p(z|x_i)$ for each x_i. To ensure an excess risk of ϵ, both N and M need be at least as large as $\mathcal{O}(1/\epsilon^2)$, making the overall samples required to be $\mathcal{O}(1/\epsilon^4)$; see (Nemirovski et al., 2009; Wang et al., 2014) and references therein. Hence, when M is small, SAA would provide poor results.

A second option is to resort to stochastic gradient methods (SGD). One can construct a *biased* stochastic estimate of the gradient using $\nabla_f L = \nabla\ell(y, \langle f, \tilde{\psi}(x)\rangle)\tilde{\psi}(x)$, where $\tilde{\psi}(x)$ is an estimate of $\mathbb{E}_{z|x}[\psi(z,x)]$ for any x. To ensure convergence, the bias of the stochastic gradient must be small, *i.e.*, a large amount of samples from each conditional distribution is needed.

Another commonly used approach is first representing the conditional distributions as the so-called kernel conditional embedding, and then performing a supervised learning step on the embedded conditional distributions (Song et al., 2013; Grunewalder et al., 2012a). This two-step procedure suffers from poor statistical sample complexity and computational cost. The kernel conditional embedding estimation costs $O(N^3)$, where N is number of pair of samples (x,z). To achieve ϵ error in the conditional kernel embedding estimation, N needs to be $\mathcal{O}(1/\epsilon^4)$[1].

Recently, Wang et al. (2014) solved a related but fundamentally distinct problem of the form,

$$\min_{f\in\mathcal{F}} L(f) = \mathbb{E}_y\left[\ell(y, \mathbb{E}_z[f(z)])\right] \quad (4)$$

where z is independent of y, and $f(z)$ is a smooth function parameterized by some finite-dimensional parameter. The authors provide an algorithm that combines stochastic gradient descent with moving average estimation for the inner expectation, and achieves an overall $\mathcal{O}(1/\epsilon^{3.5})$ sample complexity for smooth convex loss functions. The algorithm cannot directly handle random variable z with *infinite support*. Hence, such an algorithm does not apply to the more general and difficult situation that we consider in this paper.

Our approach and contribution. To address the above challenges, we propose a novel approach called *dual embeddings*. The key idea is to reformulate (1) into a min-max or saddle point problem by utilizing the Fenchel duality of the loss function. We observe that with a smooth loss function and continuous conditional distributions, the dual variables form a continuous function of x and y. Therefore, we can parameterize it as a function in some RKHS induced by any universal kernel, where the information about the marginal distribution $p(x)$ and conditional distribution $p(z|x)$ can be aggregated via a kernel embedding of the joint distribution $p(x,z)$. Furthermore, we propose an efficient algorithm based on stochastic approximation to solve the resulted saddle point problem over RKHSs, and establish finite-sample analysis of the generic learning from conditional distributions problems.

Compared to previous applicable approaches, an advantage of the proposed method is that it requires only *one sample* from each conditional distribution. Under mild conditions, the overall sample complexity reduces to $\mathcal{O}(1/\epsilon^2)$ in contrast to the $\mathcal{O}(1/\epsilon^4)$ complexity required by SAA or kernel conditional embedding. As a by-product, even in the degenerate case (4), this implies an $\mathcal{O}(1/\epsilon^2)$ sample complexity when the inner function is linear, which already surpasses the result obtained in (Wang et al., 2014) and is known to be unimprovable. Furthermore, our algorithm is generic for the family of problems of learning from conditional distributions, and can be adapted to problems with different loss functions and hypothesis function spaces.

[1] With appropriate assumptions on the joint distribution $p(x,z)$, a better rate can be obtained (Grunewalder et al., 2012a). However, for fair comparison, we did not introduce such extra assumptions.

Our proposed method also offers some new insights into several related applications. In the reinforcement learning settings, our method provides the first algorithm that truly minimizes the mean-square Bellman error (MSBE) with both theoretical guarantees and sample efficiency. We show that the existing gradient-TD2 algorithm by Sutton et al. (2009); Liu et al. (2015) is a special case of our algorithm, and the residual gradient algorithm (Baird, 1995) is derived by optimizing an upper bound of the MSBE. In the invariance learning setting, our method also provides a unified view of several existing methods for encoding invariance. Finally, numerical experiments on both synthetic and real-world datasets show that our method can significantly improve over the previous state-of-the-art performances.

2 Preliminaries

We first introduce our notation for kernels and kernel embeddings. Let $\mathcal{X} \subset \mathbb{R}^d$ be some input space and $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ be a positive definite kernel function. For notation simplicity, we denote the feature map of kernel k or $\tilde{k}$ as

$$\phi(x) := k(x, \cdot), \quad \psi(z) := \tilde{k}(z, \cdot),$$

and use $k(x, \cdot)$ and $\phi(x)$, or $\tilde{k}(z, \cdot)$ and $\psi(z)$ interchangeably. Then k induces a RKHS $\mathcal{H}$, which has the property $h(x) = \langle h, \phi(x)\rangle_{\mathcal{H}}$, $\forall h \in \mathcal{H}$, where $\langle \cdot, \cdot \rangle_{\mathcal{H}}$ is the inner product and $\|h\|^2_{\mathcal{H}} := \langle h, h\rangle_{\mathcal{H}}$ is the norm in $\mathcal{H}$. We denote all continuous functions on $\mathcal{X}$ as $\mathcal{C}(\mathcal{X})$ and $\|\cdot\|_\infty$ as the maximum norm. We call k a *universal kernel* if $\mathcal{H}$ is dense in $\mathcal{C}(\Omega')$ for any compact set $\Omega' \subseteq \mathcal{X}$, *i.e.*, for any $\epsilon > 0$ and $u \in \mathcal{C}(\Omega')$, there exists $h \in \mathcal{H}$, such that $\|u - h\|_\infty \leqslant \epsilon$.

Convex conjugate and Fenchel duality. Let $\ell : \mathbb{R}^d \to \mathbb{R}$, its convex conjugate function is defined as

$$\ell^*(u) = \sup_{v \in \mathbb{R}^d} \left(u^\top v - \ell(v)\right).$$

When ℓ is proper, convex and lower semicontinuous for any u, its conjugate function is also proper, convex and lower semicontinuous. More improtantly, the (ℓ, ℓ^*) are dual to each other, *i.e.*, $(\ell^*)^* = \ell$, which is known as Fenchel duality (Hiriart-Urruty and Lemaréchal, 2012; Rifkin and Lippert, 2007). Therefore, we can represent the ℓ by its convex conjugate as

$$\ell(v) = \sup_{u \in \mathbb{R}^d} \left(v^\top u - \ell^*(u)\right).$$

It can be shown that the supremum is achieved if $v \in \partial \ell^*(u)$, or equivalently $u \in \partial \ell(v)$.

Function approximation using RKHS. Let $\mathcal{H}^\delta := \{h \in \mathcal{H} : \|h\|^2_{\mathcal{H}} \leqslant \delta\}$ be a bounded ball in the RKHS, and we define the approximation error of the RKHS $\mathcal{H}^\delta$ as approximating continuous functions in $\mathcal{C}(\mathcal{X})$ by a function $h \in \mathcal{H}^\delta$ as (Bach, 2014; Barron, 1993)

$$\mathcal{E}(\delta) := \sup_{u \in \mathbb{C}(\mathcal{X})} \inf_{h \in \mathcal{H}^\delta} \|u - h\|_\infty. \quad (5)$$

One can immediately see that $\mathcal{E}(\delta)$ decreases as δ increases and vanishes to zero as δ goes to infinity. If $\mathcal{C}(\mathcal{X})$ is restricted to the set of uniformly bounded continuous functions, then $\mathcal{E}(\delta)$ is also bounded. The approximation property, *i.e.*, dependence on δ remains an open question for general RKHS, but has been carefully established for special kernels. For example, with the kernel $k(x, x') = 1/(1 + \exp(\langle x, x'\rangle))$ induced by the sigmoidal activation function, we have $\mathcal{E}(\delta) = O(\delta^{-2/(d+1)} \log(\delta))$ for a Lipschitz continuous function space $\mathcal{C}(\mathcal{X})$ (Bach, 2014).[2]

Hilbert space embedding of distributions. Hilbert space embeddings of distributions (Smola et al., 2007) are mappings of distributions into potentially *infinite* dimensional feature spaces,

$$\mu_x := \mathbb{E}_x[\phi(x)] = \int_{\mathcal{X}} \phi(x) p(x) dx : \mathcal{P} \mapsto \mathcal{H} \quad (6)$$

where the distribution is mapped to its expected feature map, *i.e.*, to a point in the feature space. Kernel embedding of distributions has rich representational power. Some feature maps can make the mapping injective (Sriperumbudur et al., 2008), meaning that if two distributions, $p(X)$ and $q(X)$, are different, they are mapped to two distinct points in the feature space. For instance, when $\mathcal{X} \subseteq \mathbb{R}^d$, the feature spaces of many commonly used kernels, such as the Gaussian RBF kernel, will generate injective embedding. We can also embed the joint distribution $p(x, y)$ over a pair of variables using two kernels $k(x, x) = \langle \phi(x), \phi(x')\rangle_{\mathcal{H}}$ and $\tilde{k}(z, z') = \langle \psi(z), \psi(z')\rangle_{\mathcal{G}}$ as

$$\begin{aligned} \mathcal{C}_{zx} &:= \mathbb{E}_{zx}[\psi(z) \otimes \phi(x)] \\ &= \int_{\mathcal{Z} \times \mathcal{X}} \psi(z) \otimes \phi(x) p(z, x) dz dx : \mathcal{P} \mapsto \mathcal{H} \otimes \mathcal{G}, \end{aligned}$$

where the joint distribution is mapped to a point in a tensor product feature space. Based on the embedding of joint distributions, kernel embedding of conditional distributions can be defined as $\mathcal{U}_{z|x} := \mathcal{C}_{zx}\mathcal{C}_{xx}^{-1}$ as an operator $\mathcal{H} \mapsto \mathcal{G}$ (Song et al., 2013). With $\mathcal{U}_{z|x}$, we can obtain the expectations easily, *i.e.*,

$$\mathbb{E}_{z|x}[g(z)] = \langle g, \langle \mathcal{U}_{z|x}, \phi(x)\rangle_{\mathcal{H}}\rangle_{\mathcal{G}}. \quad (7)$$

Given *i.i.d.* samples $\{(x_i, z_i)\}_{i=1}^N$ from $p(z|x)$, the estimation of $\mathcal{U}_{z|x}$ involves inverse of kernel matrix, therefore, requires computational cost $O(N^3)$.

3 The Dual Embedding Framework

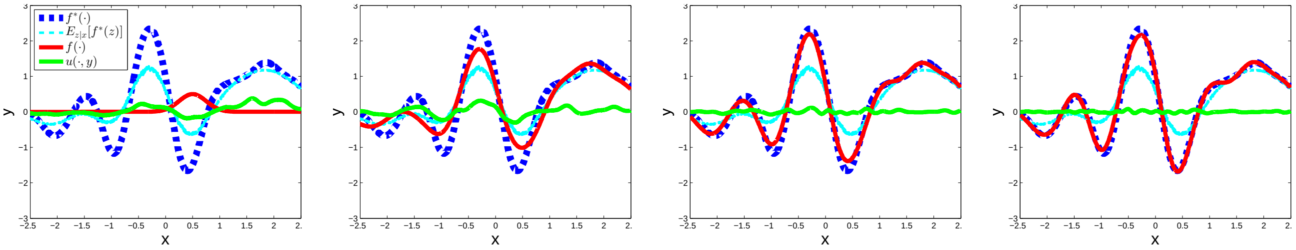
In this section, we propose a novel and sample-efficient framework to solve problem (1). Our framework leverages Fenchel duality and feature space embedding techniques to bypass the difficulties of nested expectation and the need for large sets of samples from conditional distributions. We start by introducing the interchangeability principle, which plays a fundamental role in our method.

Lemma 1 (interchangeability principle) *Let ξ be a random variable on Ξ and assume for any $\xi \in \Xi$, function $g(\cdot, \xi) : \mathbb{R} \to (-\infty, +\infty)$ is a proper and upper semicontinuous concave function. Then*

$$\mathbb{E}_\xi[\max_{u \in \mathbb{R}} g(u, \xi)] = \max_{u(\cdot) \in \mathcal{G}(\Xi)} \mathbb{E}_\xi[g(u(\xi), \xi)].$$

where $\mathcal{G}(\Xi) = \{u(\cdot) : \Xi \to \mathbb{R}\}$ is the entire space of

[2] The rate is also known to be unimprovable by DeVore et al. (1989).



(a) 0-th Iteration (b) 50-th Iteration (c) 400-th Iteration (d) 2000-th Iteration

Figure 1: Toy example with f^* sampled from a Gaussian processes. The y at position x is obtained by smoothing f^* with a Gaussian distribution condition on location x, *i.e.*, $y = \mathbb{E}_{z|x}[f^*(z)]$ where $z \sim p(z|x) = \mathcal{N}(x, 0.3)$. Given samples $\{x, y\}$, the task is to recover $f^*(\cdot)$. The blue dash curve is the ground-truth $f^*(\cdot)$. The cyan curve is the observed noisy y. The red curve is the recovered signal $f(\cdot)$ and the green curve denotes the dual function $u(\cdot, y)$ with the observed y plugged for each corresponding position x. Indeed, the dual function $u(\cdot, y)$ emphasizes the difference between y and $\mathbb{E}_{z|x}[f(z)]$ on every x. The interaction between primal $f(\cdot)$ and dual $u(\cdot, y)$ results in the recovery of the denoised signal.

functions defined on support Ξ.

The result implies that one can replace the expected value of point-wise optima by the optimum value over a function space. For the proof of lemma 1, please refer to Appendix A. More general results of interchangeability between maximization and integration can be found in (Rockafellar and Wets, 1998, Chapter 14) and (Shapiro and Dentcheva, 2014, Chapter 7).

3.1 Saddle Point Reformulation

Let the loss function $\ell_y(\cdot) := \ell(y, \cdot)$ in (1) be a proper, convex and lower semicontinuous for any y. We denote $\ell_y^*(\cdot)$ as the convex conjugate; hence $\ell_y(v) = \max_u\{uv - \ell_y^*(u)\}$, which is also a proper, convex and lower semicontinuous function. Using the Fenchel duality, we can reformulate problem (1) as

$$\min_{f\in\mathcal{F}} \mathbb{E}_{xy}\left[\max_{u\in\mathbb{R}}\left[\mathbb{E}_{z|x}[f(z,x)]\cdot u - \ell_y^*(u)\right]\right], \qquad (8)$$

Note that by the concavity and upper-semicontinuity of $-\ell_y^*(\cdot)$, for any given pair (x, y), the corresponding maximizer of the inner function always exists. Based on the interchangeability principle stated in Lemma 1, we can further rewrite (8) as

$$\min_{f\in\mathcal{F}} \max_{u(\cdot)\in\mathcal{G}(\Xi)} \Phi(f,u) := \qquad (9)$$
$$\mathbb{E}_{zxy}[f(z,x)\cdot u(x,y)] - \mathbb{E}_{xy}[\ell_y^*(u(x,y))],$$

where $\Xi = \mathcal{X} \times \mathcal{Y}$ and $\mathcal{G}(\Xi) = \{u(\cdot) : \Xi \to \mathbb{R}\}$ is the entire function space on Ξ. We emphasize that the max-operator in (8) and (9) have different meanings: the one in (8) is taken over a single variable, while the other one in (9) is over all possible function $u(\cdot) \in \mathcal{G}(\Xi)$.

Now that we have eliminated the nested expectation in the problem of interest, and converted it into a stochastic saddle point problem with an additional dual function space to optimize over. By definition, $\Phi(f, u)$ is always concave in u for any fixed f. Since $f(z, x) = \langle f, \psi(z, x)\rangle$, $\Phi(f, u)$ is also convex in f for any fixed u. Our reformulation (9) is indeed a convex-concave saddle point problem.

An example. Let us illustrate this through a concrete example. Let $f^*(\cdot) \in \mathcal{F}$ be the true function, and output $y = \mathbb{E}_{z|x}[f^*(z)]$ given x. We can recover the true function $f^*(\cdot)$ by solving the optimization problem

$$\min_{f\in\mathcal{F}} \mathbb{E}_{xy}\left[\frac{1}{2}\left(y - \mathbb{E}_{z|x}[f(z)]\right)^2\right].$$

In this example, $\ell_y(v) = \frac{1}{2}(y-v)^2$ and $\ell_y^*(u) = uy + \frac{1}{2}u^2$. Invoking the saddle point reformulation, this leads to

$$\min_{f\in\mathcal{F}} \max_{u\in\mathcal{G}(\Xi)} \mathbb{E}_{xyz}\left[(f(z) - y)\,u(x,y)\right] - \frac{1}{2}\mathbb{E}_{xy}\left[u(x,y)^2\right]$$

where the dual function $u(x, y)$ fits the discrepancy between y and $\mathbb{E}_{z|x}[f(z)]$, and thus, promotes the performance of primal function by emphasizing the different positions. See Figure 1 for the illustration of the interaction between the primal and dual functions.

3.2 Dual Continuation

Although the reformulation in (9) reveals more structure of the problem, it is not yet tractable in general. This is because the dual function $u(\cdot)$ can be an arbitrary function which we do not know how to represent. In the following, we will introduce a tractable representation for (9).

First, we define the function $u^*(\cdot) : \Xi = \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ as the *optimal dual function* if for any pair $(x, y) \in \Xi$,

$$u^*(x,y) \in \operatorname{argmax}_{u\in\mathbb{R}}\left\{u\cdot\mathbb{E}_{z|x}[f(z,x)] - \ell_y^*(u)\right\}.$$

Note the optimal dual function is well-defined since the optimal set is nonempty. Furthermore, $u^*(x, y)$ is related to the conditional distribution via $u^*(x, y) \in \partial\ell_y(\mathbb{E}_{z|x}[f(z, x)])$. This can be simply derived from convexity of loss function and Fenchel's inequality; see (Hiriart-Urruty and Lemaréchal, 2012) for a more formal argument. Depending on the property of the loss function $\ell_y(v)$, we can further derive that (see proofs in Appendix A):

Proposition 1 *Suppose both $f(z, x)$ and $p(z|x)$ are continuous in x for any z,*

(1) (Discrete case) If the loss function $\ell_y(v)$ is continuously differentiable in v for any $y \in \mathcal{Y}$, then $u^*(x, y)$

is unique and continuous in x for any $y \in \mathcal{Y}$;

(2) *(Continuous case) If the loss function $\ell_y(v)$ is continuously differentiable in (v, y), then $u^*(x, y)$ is unique and continuous in (x, y) on $\mathcal{X} \times \mathcal{Y}$.*

This assumption is satisfied widely in real-world applications. For instance, when it comes to the policy evaluation problem in 3, the corresponding optimal dual function is continuous as long as the reward function is continuous, which is true for many reinforcement learning tasks.

The fact that the optimal dual function is a continuous function has interesting consequences. As we mentioned earlier, the space of dual functions can be arbitrary and difficult to represent. Now we can simply restrict the parametrization to the space of continuous functions, which is tractable and still contains the global optimum of the optimization problem in (9). This also provides us the basis for using an RKHS to approximate these dual functions, and simply optimizing over the RKHS.

3.3 Feature Space Embedding

In the rest of the paper, we assume conditions described in Proposition 1 always hold. For the sake of simplicity, we focus on the case when $\mathcal{Y}$ is a continuous set[3]. Hence, from Proposition 1, the optimal dual function is indeed continuous in $(x, y) \in \Xi = \mathcal{X} \times \mathcal{Y}$. As an immediate consequence, we lose nothing by restricting the dual function space $\mathcal{G}(\Xi)$ to be continuous function space on Ξ. Recall that with universal kernels, we can approximate any continuous function with arbitrarily small error. Thus we approximate the dual space $\mathcal{G}(\Xi)$ by the bounded RKHS $\mathcal{H}^\delta$ induced by a universal kernel $k((x, y), (x', y')) = \langle \phi(x, y), \phi(x', y') \rangle_{\mathcal{H}}$. Therefore, $u(x, y) = \langle u, \phi(x, y) \rangle_{\mathcal{H}}$. To distinguish inner product between the primal function space $\mathcal{F}$ and the dual RKHS $\mathcal{H}^\delta$, we denote the inner product in $\mathcal{F}$ as $\langle \cdot, \cdot \rangle_{\mathcal{F}}$.

We can rewrite the objective of the saddle point problem in (9) as

$$\begin{aligned}\Phi(f, u) =&\mathbb{E}_{xyz}\left[\langle f, \psi(z, x)\rangle_{\mathcal{F}} \langle u, \phi(x, y)\rangle_{\mathcal{H}}\right.\\ &\left.- \ell_y^*(\langle u, \phi(x, y)\rangle_{\mathcal{H}})\right] \qquad (10)\\ =&f^\top \mathcal{C}_{zxy} u - \mathbb{E}_{xy}[\ell_y^*(\langle u, \phi(x, y)\rangle_{\mathcal{H}})],\end{aligned}$$

where $f(z, x) = \langle f, \psi(z, x)\rangle_{\mathcal{F}}$ by the definition of $\mathcal{F}$, and $\mathcal{C}_{zxy} = \mathbb{E}_{zxy}[\psi(z, x) \otimes \phi(x, y)]$ is the joint embedding of $p(z, x, y)$ over $\mathcal{F} \times \mathcal{H}$. The new saddle point approximation (10) based on dual kernel embedding allows us to efficiently represent the dual function and bypass the fundamental difficulty with insufficient samples from the conditional distributions. There is no need to access either the conditional distribution $p(z|x)$, the conditional expectation $\mathbb{E}_{z|x}[\cdot]$, or the conditional embedding operator $\mathcal{U}_{z|x}$ anymore, which reduces both the statistical and computational complexity.

Algorithm 1 Embedding-SGD for Optimization (10)

```
Input: $p(x,y)$, $p(z|x)$, $\psi(z,x)$, $\phi(x,y)$, $\{\gamma_i \geqslant 0\}_{i=1}^t$
Initialize $f_0$ and $u_0$ randomly.
for $i = 1, \ldots, t$ do
    Sample $(x_i, y_i) \sim p(x, y)$ and $z_i \sim p(z|x)$.
    $f_{i+1} = \Pi_{\mathcal{F}}(f_i - \gamma_i \psi(z_i, x_i) u_i(x_i, y_i))$.
    $u_{i+1} = \Pi_{\mathcal{H}^\delta}(u_i + \gamma_i[f_i(z_i, x_i) - \nabla \ell_{y_i}^*(u_i(x_i, y_i))]\phi(x_i, y_i))$
end for
Output: $\bar{f}_t = \frac{\sum_{i=1}^t \gamma_i f_i}{\sum_{i=1}^t \gamma_i}$, $\bar{u}_t = \frac{\sum_{i=1}^t \gamma_i u_i}{\sum_{i=1}^t \gamma_i}$
```

Specifically, given a set of samples (x, y, z), where $(x, y) \sim p(x, y)$ and $z \sim p(z|x)$, we can now construct an unbiased stochastic estimate for the gradient, namely,

$$\begin{aligned}\nabla_f \hat{\Phi}_{x,y,z}(f, u) &= \psi(z, x) u(x, y),\\ \nabla_u \hat{\Phi}_{x,y,z}(f, u) &= [f(z, x) - \nabla \ell_y^*(u(x, y))]\phi(x, y),\end{aligned}$$

with $\mathbb{E}\left[\nabla \hat{\Phi}_{x,y,z}(f, u)\right] = \nabla \Phi(f, u)$, respectively. For simplicity of notation, we use ∇ to denote the gradient as well as the subgradient. With the unbiased stochastic gradient, we are now able to solve the approximation problem (10) by resorting to the powerful mirror descent stochastic approximation framework (Nemirovski et al., 2009).

3.4 Sample-Efficient Algorithm

The algorithm is summarized in Algorithm 1. At each iteration, the algorithm performs a projected gradient step both for the primal variable f and dual variable u based on the unbiased stochastic gradient. The proposed algorithm avoids the need for overwhelmingly large sample sizes from the conditional distributions when estimating the gradient. At each iteration, only one sample from the conditional distribution is required in our algorithm!

Throughout our discussion, we make the following standard assumptions:

Assumption 1 *There exists constant scalars $C_{\mathcal{F}}$, $M_{\mathcal{F}}$, and c_ℓ, such that for any $f \in \mathcal{F}, u \in \mathcal{H}^\delta$,*

$$\mathbb{E}_{z,x}[\|f(z, x)\|_2^2] \leqslant M_{\mathcal{F}}, \quad \mathbb{E}_{z,x}[\|\psi(z, x)\|_{\mathcal{F}}^2] \leqslant C_{\mathcal{F}},$$
$$\mathbb{E}_y[\|\nabla \ell_y^*(u)\|_2^2] \leqslant c_\ell.$$

Assumption 2 *There exists constant $\kappa > 0$ such that $k(w, w') \leqslant \kappa$ for any $w, w' \in \mathcal{X}$.*

Assumption 1 and 2 basically suggest that the variance of our stochastic gradient estimate is always bounded. Note that we do not assume any strongly convexity/concavity or Lipschitz smoothness of the saddle point problem. Hence, we set the output as the average of intermediate solutions weighted by the learning rates $\{\gamma_i\}$, as often used in the literature, to ensure the convergence of the algorithm.

Define the accuracy of any candidate solution $(\bar{f}, \bar{u})$ to the saddle point problem as

$$\epsilon_{\text{gap}}(\bar{f}, \bar{u}) := \max_{u \in \mathcal{H}^\delta} \Phi(\bar{f}, u) - \min_{f \in \mathcal{F}} \Phi(f, \bar{u}). \tag{11}$$

We have the following convergence result,

[3] The same derivation is also applicable to discrete $\mathcal{Y}$ with $\{u_y(x)\}_{y \in \mathcal{Y}}$.

Theorem 1 *Under Assumptions 1 and 2, the solution $(\bar{f}_t, \bar{u}_t)$ after t steps of the algorithm with step-sizes being $\gamma_t = \frac{\gamma}{\sqrt{t}} (\gamma > 0)$ satisfies:*

$$\mathbb{E}[\epsilon_{\text{gap}}(\bar{f}_t, \bar{u}_t)] \leqslant [(2D_{\mathcal{F}}^2 + 4\delta)/\gamma + \gamma \mathcal{C}(\delta, \kappa)] \frac{1}{\sqrt{t}} \quad (12)$$

where $D_{\mathcal{F}}^2 = \sup_{f \in \mathcal{F}} \frac{1}{2} \|f_0 - f\|_2^2$ and $\mathcal{C}(\delta, \kappa) = \kappa(5M_{\mathcal{F}} + c_\ell) + \frac{1}{8}(\delta + \kappa)^2 C_{\mathcal{F}}$.

The above theorem implies that our algorithm achieves an overall $\mathcal{O}(1/\sqrt{t})$ convergence rate, which is known to be unimprovable already for traditional stochastic optimization with general convex loss function (Nemirovski et al., 2009). Note that in principle and in practice, we can also exploit the mini-batch trick to reduce the variance of the stochastic gradient; and this could improve the convergence up to a constant.

With the rate of the convergence of Algorithm 1 in theorem 1, let f_* be the optimal solution to (1), we further achieve the conclusion that

Corollary 1 *If $f \in \mathcal{F}$ is uniformly bounded and $\ell_y^*(v)$ is uniformly Lipschitz continuous in v for any y, then, under Assumptions 1 an 2, after t steps with $\gamma = \mathcal{O}\left(\frac{1}{\sqrt{\delta}}\right)$, the algorithm provides $\bar{f}_t$ satisfies*

$$\mathbb{E}[L(\bar{f}_t) - L(f_*)] \leqslant \mathcal{O}\left(\frac{\delta^{3/2}}{\sqrt{t}} + \mathcal{E}(\delta)\right). \quad (13)$$

There is clearly a delicate trade-off between the optimization error and approximation error. Using large δ will increase the optimization error but decrease the approximation error. When δ is moderately large (which is expected in the situation when the optimal dual function has small magnitude), our dual kernel embedding algorithm can achieve an overall $\mathcal{O}(1/\epsilon^2)$ sample complexity when solving learning problems in the form of (1). For the analysis details, please refer to Appendix C.

4 Applications

In this section, we discuss in detail how the dual embedding can be applied to solve two important learning problems, *i.e.*, learning with invariance and policy evaluation in reinforcement learning, which are special cases of the optimization in (1) and satisfy the assumptions for the convergence of our algorithm. We tailor the proposed algorithm for the respective learning scenarios and unify several existing algorithms for each learning problem into our framework. Due to the space limit, we focus only on algorithms with kernel embeddings. Extended algorithms with random feature, doubly SGD (Dai et al., 2014), neural networks as well as their hybrid can be found in Appendix E.

4.1 Learning with Invariant Representations

Invariance learning. The goal is to solve the optimization (2), which learns a function in RKHS $\tilde{\mathcal{H}}$ with kernel $\tilde{k}$. Applying the dual kernel embedding, we end up solving the saddle point problem

$$\min_{f \in \tilde{\mathcal{H}}} \max_{u \in \mathcal{H}} \; \mathbb{E}_{zx}\left[\langle f, \psi(z)\rangle_{\tilde{\mathcal{H}}} \cdot u(x)\right] \quad (14)$$
$$-\mathbb{E}_{xy}[\ell_y^*(u(x))] + \frac{\nu}{2}\|f\|_{\tilde{\mathcal{H}}}^2,$$

where $\mathcal{H}$ is the dual RKHS with the universal kernel introduced in our method.

Remark. The proposed algorithm bears some similarities to virtual sample techniques (Niyogi et al., 1998; Loosli et al., 2007) in the sense that they both create examples with prior knowledge to incorporate invariance. In fact, the virtual sample technique can be viewed as optimizing an upper bound of the objective (2) by simply moving the conditional expectation outside, *i.e.*, $\mathbb{E}_{x,y}[\ell(y, \mathbb{E}_{z|x}[f(z)])] \leqslant \mathbb{E}_{x,y,z|x}\left[\ell(y, f(z))\right]$, where the inequality comes from the convexity of $\ell(y, \cdot)$.

Remark. The learning problem (2) can be understood as learning with RKHS $\hat{\mathcal{H}}$ with Haar-Integral kernel $\hat{k}$ which is generated by $\tilde{k}$ as $\hat{k}(x, x') = \langle \mathbb{E}_{p(z|x)}[\psi(z)], \mathbb{E}_{p(z'|x')}[\psi(z')]\rangle_{\tilde{\mathcal{H}}}$, with implicit feature map $\mathbb{E}_{p(z|x)}[\psi(z)]$. If $f \in \tilde{\mathcal{H}}$, then, $f(x) = \mathbb{E}_{z|x}[\langle f, \psi(z)\rangle_{\tilde{\mathcal{H}}}] = \langle f, \mathbb{E}_{z|x}[\psi(z)]\rangle \in \hat{\mathcal{H}}$. The Haar-Integral kernel can be viewed as a special case of Hilbertian metric on probability measures on which the output of function should be invariant (Hein and Bousquet, 2005). Therefore, other kernels defined for distributions, *e.g.*, the probability product kernel (Jebara et al., 2004), can also be used in incorporating invariance.

Remark. Robust learning with contamined samples can also be viewed as incorporating an invariance prior w.r.t. the perturbation distribution into learning procedure. Therefore, rather than resorting to robust optimization techniques (Bhattacharyya et al., 2005; Ben-Tal and Nemirovski, 2008), the proposed algorithm for learning with invariance serves as a viable alternative for robust learning.

4.2 Reinforcement Learning

Policy evaluation. The goal is to estimate the value function $V^\pi(\cdot)$ of a given policy $\pi(a|s)$ by minimizing the mean-square Bellman error (MSBE) (3).

With $V^\pi \in \tilde{\mathcal{H}}$ with feature map $\psi(\cdot)$, *i.e.*, $V^\pi(s) = \langle V^\pi, \psi(s)\rangle_{\tilde{\mathcal{H}}}$, this optimization is clearly a special case of (1) as:

$$\min_{V^\pi \in \tilde{\mathcal{H}}} \mathbb{E}_{s,a}\left[\left(R(s,a) - \mathbb{E}_{s'|a,s}\left[\langle V^\pi, \psi(s) - \gamma\psi(s')\rangle\right]^2\right)\right]. \quad (15)$$

Applying the dual kernel embedding, we end up solving the saddle point problem

$$\min_{V^\pi \in \tilde{\mathcal{H}}} \max_{u \in \mathcal{H}} \mathbb{E}_{s',a,s}\Big[\,(R(s,a) - \langle V^\pi, \psi(s) - \gamma\psi(s')\rangle_{\tilde{\mathcal{H}}})\, u(s)\Big]$$
$$-\frac{1}{2}\mathbb{E}_s[u^2(s)]. \quad (16)$$

Remark. The algorithm can be extended to off-policy setting via the adjusted objective with importance ratio between current policy and the behavior policy.

Remark. We used different RKHSs for primal and dual functions. If we use the *same finite basis functions* to parametrize both the value function and the dual function, *i.e.*, $V^{\pi}(s) = \theta^T\psi(s)$ and $u(s) = \eta^T\psi(s)$, where $\psi(s) = [\psi_i(z)]_{i=1}^d \in \mathbb{R}^d$, $\theta, \eta \in \mathbb{R}^d$, our saddle point problem (16) reduces to $\min_\theta \left\|\mathbb{E}_{s,a,s'}[\Delta_\theta(s,a,s')\psi]\right\|^2_{\mathbb{E}[\psi\psi^\top]^{-1}}$, where $\Delta_\theta(s,a,s') = R(s,a) + \gamma V^{\pi}(s') - V^{\pi}(s)$. This is exactly the same as the objective proposed in (Sutton et al., 2009) of gradient-TD2. Moreover, the update rules in gradient-TD2 can also be derived by conducting the proposed Embedding-SGD with such parametrization. For details of the derivation, please refer to Appendix D.

From this perspective, gradient-TD2 is simply a special case of the proposed Embedding-SGD applied to policy evaluation with particular parametrization. However, in the view of our framework, there is really no need to restrict to the same finite parametric model for the value and dual functions. As further demonstrated in our experiments, with different nonparametric models, the performances can be improved significantly. See details in Section 5.2.

The residual gradient (Baird, 1995) is applying stochastic gradient descent to $\mathbb{E}_{s,a,s'}\left[\Delta_\theta(s,a,s')^2\right]$ with parametric form $V^{\pi}(s) = \theta^T\psi(s)$. Indeed, this objective is an upper bound of MSBE (3) because of the convexity of square loss.

Our algorithm is also fundamentally different from the TD algorithm even in the finite state case. The TD algorithm updates the state-value function directly by an estimate of the temporal difference based on one pair of samples, while our algorithm updates the state-value function based on accumulated estimate of the temporal difference, which intuitively is more robust.

5 Experiments

We test the proposed algorithm on two applications, *i.e.*, learning with invariant representation and policy evaluation. For full details of our experimental setups, please refer to Appendix F.

5.1 Experiments on Invariance Learning

To justify the algorithm for learning with invariance, we test the algorithm on two tasks. We first apply the algorithm to robust learning problem where the inputs are contaminated, and then, we conduct comparison on a molecular energetics prediction problem (Montavon et al., 2012). We compare the proposed algorithm with SGD with the virtual samples technique (Niyogi et al., 1998; Loosli et al., 2007) and SGD with finite sample average for inner expectation (SGD-SAA). We use Gaussian kernels in all tasks. To demonstrate the sample-efficiency of our algorithm, 10 virtual samples are generated for each datum in the training phase. The algorithms are terminated after going through 10 rounds of the dataset. We emphasize that SGD with virtual samples is optimizing an upper bound of the objective, and thus, it is predictable that our algorithm can achieve better performance. We plot this result with a dotted line.

Noisy measurement. We generate a synthetic dataset by

$$\begin{aligned}\bar{x} &\sim \mathcal{U}([-0.5, 0.5]), \quad x = \bar{x} + 0.05e,\\ y &= (\sin(3.53\pi\bar{x}) + \cos(7.7\pi\bar{x}))\exp(-1.6\pi|\bar{x}|)\\ &+ \quad 3\bar{x}^2 + 0.01e,\end{aligned}$$

where the contamination $e \sim \mathcal{N}(0,1)$. Only (x, y) are provided to learning methods, while $\bar{x}$ is unknown. The virtual samples are sampled from $z \sim \mathcal{N}(x, 0.05^2)$ for each observation. The 10 runs average results are illustrated in Figure 2(a). The proposed algorithm achieves average MSE as low as 0.0029 after visit 0.1M data, significantly better than the alternatives.

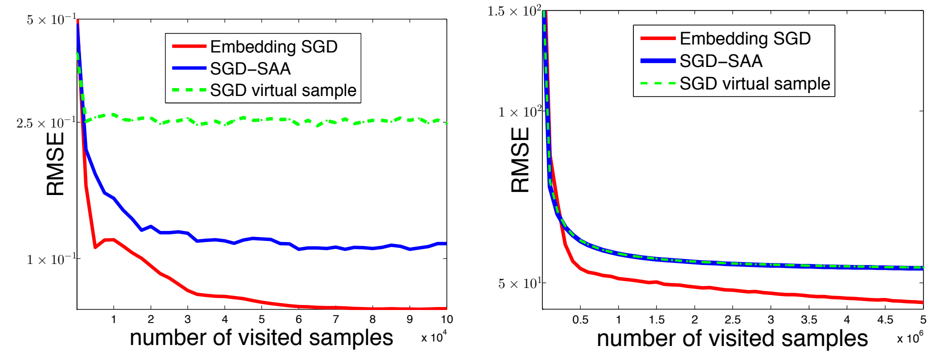
QuantumMachine. We test the proposed algorithm for learning with invariance task on the QuantumMachine 5-fold dataset for atomization energy prediction. We follow the same setting in (Montavon et al., 2012) where the data points are represented by Coulomb matrices, and the virtual samples are generated by random permutation. The average results are shown in Figure 2(b). The proposed algorithm achieves a significant better solution, while SGD-SAA and SGD with virtual samples stuck in inferior solutions due to the inaccurate inner expectation estimation and optimizing indirect objective, respectively.

5.2 Experiments on Policy Evaluation

We compare the proposed algorithm to several prevailing algorithms for policy evaluation, including gradient-TD2 (GTD2) (Sutton et al., 2009; Liu et al., 2015), residual gradient (RG) (Baird, 1995) and kernel MDP (Grunewalder et al., 2012b) in terms of mean square Bellman error (Dann et al., 2014). We should point out that kernel MDP is not an online algorithm, since it must visit the entire dataset when estimating the embedding and inner expectation in each iteration. We conduct experiments for policy evaluation on several benchmark datasets, including navigation, cart-pole swing up and PUMA-560 manipulation. We use Gaussian kernels in the nonparametric algorithms, *i.e.*, kernel MDP and Embedding SGD, while we test with random Fourier features (Rahimi and Recht, 2008) for the parametric competitors, *i.e.*, GTD2 and RG. In order to demonstrate the sample efficiency of our method, we only use one sample from the conditional distribution in the training phase, therefore, the cross-validation based on Bellman error is not appropriate. We perform a parameter sweep as (Silver et al., 2014). See appendix F for detailed settings. Results are averaged over 10 independent trials.

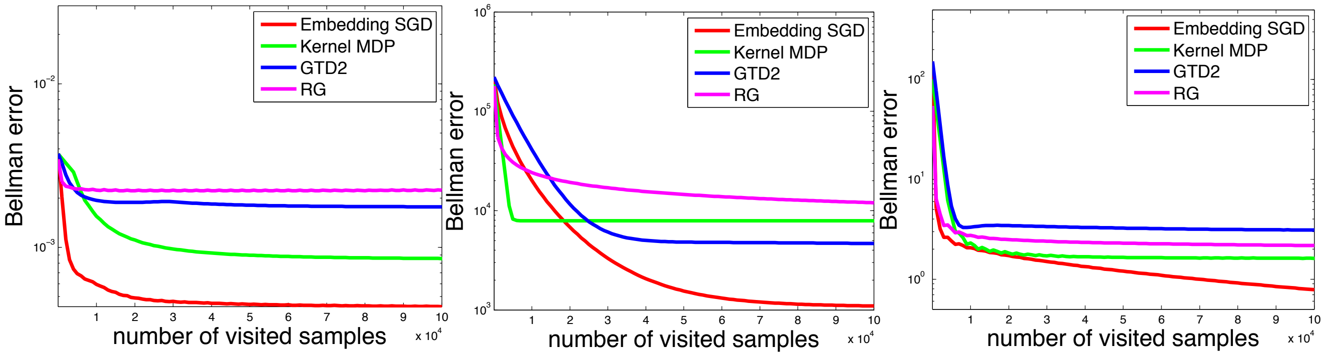
Navigation. The navigation in an unbounded room experiment extends the discretized MDP in (Grunewalder et al., 2012b) to a continuous state and action MDP. Specifically, the reward is $R(s) = \exp(-100\|s\|^2)$. We evaluate the deterministic policy $\pi(s) = -0.2sR(s)$, following the gradient of the reward function. The transition distribution follows a Gaussian distribution, $p(s'|a,s) = \mathcal{N}(s + a, 0.1I)$. Results are reported in Figure 3(a).

Cart-pole swing up. The cart-pole system consists of a cart and a pendulum. It is an under-actuated system with



(a) Robust Learning (b) QuantumMachine

Figure 2: Learning with invariance.



(a) Navigation (b) Cart-Pole (c) PUMA-560

Figure 3: Policy evaluation.

only one control action on the cart. The reward is maximized if the pendulum is swung up to π angle with zero velocity. We evaluate a linear policy $\pi(s) = As + b$ where $A \in \mathbb{R}^{1\times 4}$ and $b \in \mathbb{R}^{1\times 1}$. Results are reported in Figure 3(b).

PUMA-560 manipulation. PUMA-560 is a robotic arm that has 6 degrees of freedom with 6 actuators on each joint. The task is to steer the end-effector to the desired position and orientation with zero velocity. The reward function is maximum if the arm is located to the desired position. We evaluate a linear policy $\pi(s) = As + b$ where $A \in \mathbb{R}^{6\times 12}$ and $b \in \mathbb{R}^{6\times 1}$. Results are reported in Figure 3(c).

In all experiments, the proposed algorithm performs consistently better than the competitors. The advantages of the proposed algorithm mainly come from three aspects: **i)**, it utilizes more flexible dual function space, rather than the constrained space in GTD2; **ii)**, it directly optimizes the MSBE, rather than its surrogate as in GTD2 and RG; **iii)**, it directly targets on value function estimation and forms an one-shot algorithm, rather than a two-stage procedure in kernel MDP including estimating conditional kernel embedding as intermediate step.

6 Conclusion

We propose a novel *sample-efficient* algorithm, **Embedding-SGD**, for addressing learning from conditional distributions problems. Our algorithm benefits from a novel use of saddle point and kernel embedding techniques, to mitigate the difficulty with limited samples from conditional distribution as well as the presence of nested expectations. To our best knowledge, among all existing algorithms able to solve such problems, this is *the first* algorithm that allows us to take only one sample at a time from the conditional distribution yet comes with provable theoretical guarantees.

We apply the proposed algorithm to solve two fundamental problems in machine learning, *i.e.*, learning with invariance and policy evaluation in reinforcement learning. The proposed algorithm achieves the state-of-the-art performance on these two tasks compared with existing algorithms.

In addition to its wide applicability, our algorithm is also very versatile and can be easily extended with random features E.1 or the doubly stochastic gradient trick E.2. Moreover, we can extend the framework to the dual neural network embedding E.3. It should be emphasized that since the primal and dual function spaces are designed for different purposes, although we use RKHSs for both in the main text for simplicity, we can use different function approximators separately for primal and dual functions.

Acknowledgements. This project was supported in part by NSF IIS-1218749, NIH BIGDATA 1R01GM108341, NSF CAREER IIS-1350983, NSF IIS-1639792 EAGER, ONR N00014-15-1-2340, Nvidia and Intel.

References

Anselmi, F., Leibo, J. Z., Rosasco, L., Mutch, J., Tacchetti, A., and Poggio, T. (2013). Unsupervised learning of invariant representations in hierarchical architectures. *arXiv preprint arXiv:1311.4158*.

Bach, F. R. (2014). Breaking the curse of dimensionality with convex neural networks. *CoRR*, abs/1412.8690.

Baird, L. (1995). Residual algorithms: reinforcement learning with function approximation. In *Proc. Intl. Conf. Machine Learning*, pages 30–37. Morgan Kaufmann.

Barron, A. R. (1993). Universal approximation bounds for superpositions of a sigmoidal function. *IEEE Trans. Inform. Theory*, 39(3):930–945.

Ben-Tal, A. and Nemirovski, L. E. G. . A. (2008). *Robust Optimization*. Princeton University Press, Princeton, NJ.

Bhattacharyya, C., Pannagadatta, K. S., and Smola, A. J. (2005). A second order cone programming formulation for classifying missing data. In *Advances in Neural Information Processing Systems 17*, pages 153–160. MIT Press, Cambridge, MA.

Dai, B., Xie, B., He, N., Liang, Y., Raj, A., Balcan, M.-F., and Song. L. Scalable kernel methods via doubly stochastic gradients. In *Advances in Neural Information Processing Systems*, pages 3041–3049, 2014.

Dann, C., and Neumann, G., and Peters, J. (2014). Policy evaluation with temporal differences: a survey and comparison. In *Journal of Machine Learning Research*, pages 809–883.

Devinatz, A. (1953). Integral representation of pd functions. *Trans. AMS*, 74(1):56–77.

DeVore, R., Howard, R., and Micchelli, C. (1989). Optimal nonlinear approximation. *Manuskripta Mathematika*.

Grunewalder, S., Lever, G., Baldassarre, L., Patterson, S., Gretton, A., and Pontil, M. (2012a). Conditional mean embeddings as regressors. In *ICML*.

Grunewalder, S., Lever, G., Baldassarre, L., Pontil, M., and Gretton, A. (2012b). Modeling transition dynamics in MDPs with RKHS embeddings. In *ICML*.

Hein, M. and Bousquet, O. (2004). Kernels, associated structures, and generalizations. Technical Report 127, Max Planck Institute for Biological Cybernetics.

Hein, M. and Bousquet, O. (2005). Hilbertian metrics and positive definite kernels on probability measures. In *Proc. of AI & Statistics*, volume 10, pages 136–143.

Hiriart-Urruty, J.-B. and Lemaréchal, C. (2012). *Fundamentals of convex analysis*. Springer Science & Business Media.

Jebara, T., Kondor, R., and Howard, A. (2004). Probability product kernels. *J. Mach. Learn. Res.*, 5:819–844.

Liu, B., Liu, J., Ghavamzadeh, M., Mahadevan, S., and Petrik, M. (2015). Finite-sample analysis of proximal gradient td algorithms. In *Uncertainty in Artificial Intelligence (UAI)*. AUAI Press.

Loosli, G., Canu, S., and Bottou, L. (2007). Training invariant support vector machines with selective sampling. In *Large Scale Kernel Machines*, pages 301–320. MIT Press.

Montavon, G., Hansen, K., Fazli, S., Rupp, M., Biegler, F., Ziehe, A., Tkatchenko, A., von Lilienfeld, A., and Müller, K.-R. (2012). Learning invariant representations of molecules for atomization energy prediction. In *Neural Information Processing Systems*, pages 449–457.

Mroueh, Y., Voinea, S., and Poggio, T. A. (2015). Learning with group invariant features: A kernel perspective. In *Advances in Neural Information Processing Systems 28*, pages 1558–1566.

Nemirovski, A., Juditsky, A., Lan, G., and Shapiro, A. (2009). Robust stochastic approximation approach to stochastic programming. *SIAM J. on Optimization*, 19(4):1574–1609.

Nesterov, Y. (2005). Smooth minimization of non-smooth functions. *Math. Program.*, 103(1):127–152.

Niyogi, P., Girosi, F., and Poggio, T. (1998). Incorporating prior knowledge in machine learning by creating virtual examples. *Proceedings of IEEE*, 86(11):2196–2209.

Rahimi, A. and Recht, B. (2008). Random features for large-scale kernel machines. In *Advances in Neural Information Processing Systems 20*. MIT Press, Cambridge, MA.

Rahimi, A. and Recht, B. (2009). Weighted sums of random kitchen sinks: Replacing minimization with randomization in learning. In *Neural Information Processing Systems*.

Rifkin, R. and Lippert, R. Value regularization and Fenchel duality. *Journal of Machine Learning Research*, 8:441–479, 2007.

Rockafellar, R.T., and Wets, R.J.-B. (1988). Variational analysis. *Springer*, Berlin.

Shapiro, A., and Dentcheva, D. (2014). Lectures on stochastic programming: modeling and theory. *SIAM (Vol. 16)*.

Silver, D., Lever, G., Heess, N., Degris, T., Wierstra, D., and Riedmiller, M. (2014). Deterministic policy gradient algorithms. In *ICML*.

Smola, A. J., Gretton, A., Song, L., and Schölkopf, B. (2007). A Hilbert space embedding for distributions. In *Proceedings of the International Conference on Algorithmic Learning Theory*, volume 4754, pages 13–31. Springer.

Song, L., Gretton, A., and Fukumizu, K. (2013). Kernel embeddings of conditional distributions. *IEEE Signal Processing Magazine*.

Sriperumbudur, B., Gretton, A., Fukumizu, K., Lanckriet, G., and Schölkopf, B. (2008). Injective Hilbert space embeddings of probability measures. In *Proc. Annual Conf. Computational Learning Theory*, pages 111–122.

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

Sutton, R. S., Maei, H. R., and Szepesvri, C. (2008). A convergent $o(n)$ temporal-difference algorithm for off-policy learning with linear function approximation. In *Advances in Neural Information Processing Systems 21*, pages 1609–1616.

Wang, M., Fang, E. X., and Liu, H. (2014). Stochastic compositional gradient descent: Algorithms for minimizing compositions of expected-value functions. *arXiv preprint arXiv:1411.3803*.

Appendix

A Interchangeability Principle and Dual Continuity

Lemma 1 *Let ξ be a random variable on Ξ and assume for any $\xi \in \Xi$, function $g(\cdot, \xi) : \mathbb{R} \to (-\infty, +\infty)$ is a proper and upper semicontinuous concave function. Then*

$$\mathbb{E}_\xi[\max_{u \in \mathbb{R}} g(u, \xi)] = \max_{u(\cdot) \in \mathcal{G}(\Xi)} \mathbb{E}_\xi[g(u(\xi), \xi)].$$

where $\mathcal{G}(\Xi) = \{u(\cdot) : \Xi \to \mathbb{R}\}$ is the entire space of functions defined on support Ξ.

Proof First of all, by assumption of concavity and upper-semicontinuity, we know that for any $\xi \in \Xi$, there exists a maximizer for $\max_u g(u, \xi)$; let us denote as u^*_ξ. We can therefore define a function $u^*(\cdot) : \mathcal{X} \to \mathbb{R}$ such that $u^*(\xi) = u^*_\xi$, and thus, $u^*(\cdot) \in \mathcal{G}(\Xi)$. Hence,

$$\mathbb{E}_\xi[\max_{u \in \mathbb{R}} g(u, \xi)] = \mathbb{E}_\xi[g(u^*(\xi), \xi)] \leqslant \max_{u(\cdot) \in \mathcal{G}(\mathcal{X})} \mathbb{E}_\xi[g(u(\xi), \xi)].$$

On the other hand, clearly, for any $u(\cdot) \in \mathcal{G}(\Xi)$ and $\xi \in \Xi$, $g_\xi(u(\xi), \xi) \leqslant \max_{u \in \mathbb{R}} g(u, \xi)$. Hence, $\mathbb{E}_\xi[g(u(\xi), \xi)] \leqslant \mathbb{E}_\xi[\max_{u \in \mathbb{R}} g(u, \xi)]$, for any $u(\cdot) \in \mathcal{G}(\Xi)$. This further implies that

$$\max_{u(\cdot) \in \mathcal{G}(\Xi)} \mathbb{E}_\xi[g(u(\xi), \xi)] \leqslant \mathbb{E}_\xi[\max_{u \in \mathbb{R}} g(u, \xi)].$$

Combining these two facts leads to the statement in the lemma. ■

Proposition 1 *Suppose both $f(z, x)$ and $p(z|x)$ are continuous in x for any z,*

(1) (Discrete case) If the loss function $\ell_y(v)$ is continuously differentiable in v for any $y \in \mathcal{Y}$, then $u^*(x, y)$ is unique and continuous in x for any $y \in \mathcal{Y}$;

(2) (Continuous case) If the loss function $\ell_y(v)$ is continuously differentiable in (v, y), then $u^*(x, y)$ is unique and continuous in (x, y) on $\mathcal{X} \times \mathcal{Y}$.

Proof The continuity properties of optimal dual function follows directly from the fact that $u^*(x, y) \in \partial \ell_y(\mathbb{E}_{z|x}[f(z, x)])$. In both cases, for any $y \in \mathcal{Y}$, $\ell_y(\cdot)$ is differentiable. Hence $u^*(x, y) = \ell'_y(\int f(z, x)p(z|x)dz)$ is unique. Since $f(z, x)$ and $p(z|x)$ is continuous in x for any z, then $\mathbb{E}_{z|x}[f(z, x)]$ is continuous in x. Since for any $y \in \mathcal{Y}$, $\ell'_y(\cdot)$ is continuous, the composition $u^*(x, y)$ is therefore continuous in x as well. Moreover, if $\ell'_y(\cdot)$ is also continuous in $y \in \mathcal{Y}$, then the composition $u^*(x, y)$ is continuous in (x, y). ■

Indeed, suppose $\ell_y(\cdot)$ is uniformly L-Lipschitz differentiable for any $y \in \mathcal{Y}$, $f(z, x)$ is uniformly M_f-Lipschitz continuous in x for any z, $p(z|x)$ is M_p-Lipschitz continuous in x. Then

$$\begin{aligned}
|u^*(x_1, y) - u^*(x_2, y)| &= \left| \ell'_y \left(\int f(z, x_1)p(z|x_1)dz \right) - \ell'_y \left(\int f(z, x_2)p(z|x_2)dz \right) \right| \\
&\leqslant L \int |f(z, x_1)p(z|x_1) - f(z, x_2)p(z|x_2)|dz \\
&\leqslant L \int |f(z, x_1) - f(z, x_2)|p(z|x_1)dz + L \int |f(z, x_2)| \cdot |p(z|x_1) - p(z|x_2)|dz \\
&\leqslant LM_f|x_1 - x_2| + LM_p|x_1 - x_2| \sup_x \int |f(z, x)|dz
\end{aligned}$$

If for any $f(z, x)$ is Lebesgue integrable and $\int |f(z, x)|dz$ is uniformly bounded, then $u^*(x, y)$ is also Lipschitz-continuous for any $y \in \mathcal{Y}$. Moreover, if in addition, $\ell_y(v)$ is also Lipschitz differentiable in (v, y), then $u^*(x, y)$ is also Lipschitz continuous on $\mathcal{X} \times \mathcal{Y}$.

B Preliminaries: Stochastic Approximation for Saddle Point Problems

Consider the stochastic saddle point (min-max) problem

$$\min_{x \in X} \max_{y \in Y} \Phi(x, y) = \mathbb{E}[F(x, y, \xi)]$$

where the expected value function $f(x, y)$ is convex in x and concave in y, and domains X, Y are convex closed. Let $z = [x, y]$ and $G(z, \xi) = [\nabla F_x(x, y, \xi); -\nabla F_y(x, y, \xi)]$ be the stochastic gradient for any input point z and sample ξ. Let $\|\cdot\|$ be a norm defined on the embedding Hilbert space of $Z = X \times Y$, and $D(z, z') := w(z) - w(z') - \nabla w(z')'(z - z')$ be a Bregman distance on Z defined by a 1-strongly convex (w.r.t. the norm $\|\cdot\|$) and continuously differentiable function $w(z)$. For instance, when $w(z) = \frac{1}{2}\|z\|^2$, the Bregman distance becomes $D(z, z') = \frac{1}{2}\|z - z'\|^2$.

Mirror descent SA. The mirror descent stochastic approximation (Nemirovski et al., 2009) works as follows:

$$z_i = \operatorname*{argmin}_{z \in Z}\{D(z, z_i) + \gamma_i G(z_i, \xi_i)\}, i = 1, \ldots, t.$$

The quality of an approximate solution $\bar{z} = (\bar{x}, \bar{y})$ is defined by the error

$$\epsilon_{\text{gap}}(\bar{x}, \bar{y}) := \max_{y \in Y} \Phi(\bar{x}, y) - \Phi^* + \Phi^* - \min_{x \in X} \Phi(x, \bar{y}) = \max_{y \in Y} \Phi(\bar{x}, y) - \min_{x \in X} \Phi(x, \bar{y}),$$

where Φ^* denotes the optimal value. Let $\bar{z}_t := \frac{\sum_{i=1}^t \gamma_i z_i}{\sum_{i=1}^t \gamma_i}$, the convergence properties of this weighted averaging solution is as follows.

Lemma 2 *(Nemirovski et al., 2009) Suppose* $\mathbb{E}[\|G(z_i, \xi_i)\|_*^2] \leqslant M^2, \forall i$, *we have*

$$\mathbb{E}[\epsilon_{\text{gap}}(\bar{x}_t, \bar{y}_t)] \leqslant \frac{2\max_{z \in Z} D(z, z_1) + \frac{5}{2}M^2 \sum_{i=1}^t \gamma_i^2}{\sum_{i=1}^t \gamma_i}.$$

In particular, when $\gamma_i = \frac{\gamma}{\sqrt{t}}, \forall i = 1, \ldots, t$, we have

$$\mathbb{E}[\epsilon_{\text{gap}}(\bar{x}_t, \bar{y}_t)] \leqslant (2\max_{z \in Z} D(z, z_1)/\gamma + \frac{5}{2}M^2\gamma)\frac{1}{\sqrt{t}}.$$

Moreover, suppose $D^2 = \max_{z \in Z} D(z, z_1)$ and M^2 are known, by setting $\gamma = \frac{2D}{\sqrt{5}M}$, we further have

$$\mathbb{E}[\epsilon_{\text{gap}}(\bar{x}_t, \bar{y}_t)] \leqslant \frac{2\sqrt{5}DM}{\sqrt{t}}.$$

To summarize, the mirror descent stochastic approximation achieves an $\mathcal{O}(1/\sqrt{t})$ convergence rate (also known to be unimprovable (Nemirovski et al., 2009)). Our Embedding-SGD algorithm 1 builds upon on this framework to solve the saddle point approximation problem (10).

C Convergence Analysis for Embedding-SGD

C.1 Decomposition of generalization error

We first observe that

Proposition 2 *If $f \in \mathcal{F}$ is uniformly bounded by C and $\ell_y^*(v)$ is uniformly K-Lipschitz continuous in v for any y, then $\Phi(f, u)$ is $(C + K)$-Lipschitz continuous on $\mathcal{G}(\Xi)$ with respect to $\|\cdot\|_\infty$, i.e.*

$$|\Phi(f, u_1) - \Phi(f, u_2)| \leqslant (C + K)\|u_1 - u_2\|_\infty, \forall u_1, u_2 \in \mathcal{G}(\Xi).$$

Let f_* be the optimal solution to our objective. Denote $\hat{L}(f) = \max_{u \in \mathcal{H}^\delta} \phi(f, u)$. Invoking the Lipschitz continuity of Φ, $L(f) - \hat{L}(f) \leqslant (K + C)\mathcal{E}(\delta), \forall f$. Therefore, we can decompose the error as

$$\begin{aligned} L(\bar{f}_t) - L(f_*) =& L(\bar{f}_t) - \hat{L}(\bar{f}_t) + \hat{L}(\bar{f}_t) - \hat{L}(f_*) + \hat{L}(f_*) - L(f_*) \\ &\leqslant \epsilon_{\text{gap}}(\bar{f}_t, \bar{u}_t) + 2(K + C)\mathcal{E}(\delta). \end{aligned} \tag{17}$$

C.2 Optimization error

Proof of Theorem 1 Our proof builds on results of stochastic approximation discussed in the previous section. Let M_1 and M_2 be such that for any $f \in \{f_i\}_{i=1}^t$ and $u \in \{u_i\}_{i=1}^t$,

$$\mathbb{E}_{x,y,z}[\|\nabla_f \hat{\Phi}_{x,y,z}(f, u)\|_{\mathcal{F}}^2] \leqslant M_1^2,$$

$$\mathbb{E}_{x,y,z}[\|\nabla_u \hat{\Phi}_{x,y,z}(f, u)\|_{\mathcal{H}}^2] \leqslant M_2^2.$$

Then from Lemma 2, we have

$$\mathbb{E}[\epsilon_{\text{gap}}(\bar{f}_t,\bar{y}_t)] \leqslant \frac{2(D_{\mathcal{F}}^2+D_{\mathcal{H}}^2)+\frac{5}{2}(M_1^2+M_2^2)\sum_{i=1}^t\gamma_i^2}{\sum_{i=1}^t\gamma_i} \tag{18}$$

where $D_{\mathcal{F}}^2 = \sup_{f\in\mathcal{F}}\frac{1}{2}\|f_1-f\|_2^2$ and $D_{\mathcal{H}}^2=\sup_{u\in\mathcal{H}^\delta}\frac{1}{2}\|u_1-u\|_{\mathcal{H}}^2\leqslant 2\delta$. It remains to find upper bounds for M_1 and M_2. Note that since $\|k(w,w')\|_\infty\leqslant\kappa$ for any w and w',

$$\mathbb{E}[\|\nabla_u\hat{\Phi}_{x,y,z}(f,u)\|_{\mathcal{H}}^2] \quad \leqslant \kappa\mathbb{E}[\|f(z,x)-\nabla\ell_y^*(u(x))\|^2]\leqslant 2\kappa(M_{\mathcal{F}}+c_\ell).$$

Since $u(x)=\langle u(\cdot),k(x,\cdot)\rangle_{\mathcal{H}}$, from Young's inequality, we have $|u(x)|\leqslant\frac{1}{2}\|u\|_{\mathcal{H}}^2+\frac{1}{2}\|k(x,\cdot)\|_{\mathcal{H}}^2\leqslant\frac{1}{2}(\delta+\kappa)$, for any $w\in\mathcal{X}$.

$$\mathbb{E}_{x,y,z}[\|\nabla_f\hat{\Phi}_{x,y,z}(f,u)\|_{\mathcal{F}}^2] \quad = \mathbb{E}[\|\psi(z,x)\|_{\mathcal{F}}^2u(x)^2]\leqslant\frac{1}{4}(\delta+\kappa)^2C_{\mathcal{F}}.$$

Plugging in $M_1^2=2\kappa(M_{\mathcal{F}}+c_\ell)$ and $M_2^2=\frac{1}{4}(\delta+\kappa)^2C_{\mathcal{F}}$ to (18) and setting $\gamma_t=\gamma/\sqrt{t}$, we arrive at (12). ■

C.3 Generalization Error

Proof of Corollary 1 Combining with the approximation error and optimization error into (17), we arrive at

$$\mathbb{E}[L(\bar{f}_t)-L(f_*)]\leqslant[(2D_{\mathcal{F}}^2+4\delta)/\gamma+\gamma\mathcal{C}(\delta,\kappa)]\frac{1}{\sqrt{t}}+2(K+C)\mathcal{E}(\delta)$$

Minimizing over $\gamma>0$, we get the "theoretical" optimal choice for the γ as $\gamma^*=\sqrt{\frac{2D_{\mathcal{F}}^2+4\delta}{\mathcal{C}(\delta,\kappa)}}$. Ignoring the dependence on the other parameters except δ, $\gamma^*=\mathcal{O}(1/\sqrt{\delta})$ and this further leads to

$$\mathbb{E}[L(\bar{f}_t)-L(f_*)]\leqslant\mathcal{O}\left(\frac{\delta^{3/2}}{\sqrt{t}}+\mathcal{E}(\delta)\right). \tag{19}$$

■

D Gradient-TD2 As Special Case of Embedding-SGD

Follow the notation in section 4, with the parameterization that $V^\pi(s)=\theta^T\psi(s)$ and $u(s)=\eta^T\psi(s)$, where $\psi(s)=[\psi_i(z)]_{i=1}^d\in\mathbb{R}^d$, $\theta,\eta\in\mathbb{R}^d$, the optimization becomes

$$\min_\theta\max_\eta\widehat{\Phi}(\theta,\eta):=\mathbb{E}_s\mathbb{E}_{s',a|s}\left[\Delta_\theta(s,a,s')\psi(s)^\top\eta\right]-\frac{1}{2}\mathbb{E}_s[\eta^\top\psi(s)\psi^\top(s)\eta], \tag{20}$$

where $\Delta_\theta(s,a,s')=\big(R(s)+\gamma\theta^\top\psi(s')-\theta^\top\psi(s)\big)$. For arbitrary θ, we have the closed form of $\eta(\theta)^*$ which achieves the maximum of $\widehat{\Phi}(\theta,\eta)$. Specifically, we first take derivative of $\widehat{\Phi}(\theta,\eta)$ w.r.t. η,

$$\nabla_\eta\widehat{\Phi}(\theta,\eta)=\mathbb{E}_{s,a,s'}\left[\Delta_\theta(s,a,s')\psi(s)\right]-\mathbb{E}_s\left[\psi(s)\psi(s)^\top\eta\right], \tag{21}$$

and make the derivative equal to zero,

$$\eta(\theta)^*=\mathbb{E}_s\left[\psi(s)\psi(s)^\top\right]^{-1}\mathbb{E}_{s,a,s'}\left[\Delta_\theta(s,a,s')\psi(s)\right]. \tag{22}$$

Plug the $\eta(\theta)^*$ into $\widehat{\Phi}(\theta,\eta)$, we achieve the optimization

$$\min_\theta\mathbb{E}_{s,a,s'}\left[\Delta_\theta(s,a,s')\psi(s)^\top\right]\mathbb{E}_s\left[\psi(s)\psi(s)^\top\right]^{-1}\mathbb{E}_{s,a,s'}\left[\Delta_\theta(s,a,s')\psi(s)\right], \tag{23}$$

which is exactly the objective of gradient-TD2 (Sutton et al., 2009; Liu et al., 2015). Plug the parametrization into the proposed embedding-SGD, we will achieve the update rules in i-th iteration proposed in gradient-TD2 for θ and η as

$$\begin{array}{rcl}\eta_{i+1} & = & \eta_i+\gamma_i[\Delta_\theta(s,a,s')-u_i(s)]\psi(s),\\ \theta_{i+1} & = & \theta_i-\gamma_iu_i(s)(\gamma\psi(s')-\psi(s)).\end{array}$$

Therefore, from this perspective, gradient-TD2 is simply a special case of the proposed Embedding-SGD applied to policy evalution with particular parametrization.

E Dual Embedding with Arbitrary Funtion Approximator

In the main text, we only focus on using different RKHSs as the primal and dual function spaces. As we introduce in section 1, the proposed algorithm is versatile and can be conducted with arbitrary function space for the primal or dual functions. In this section, we demonstrate applying the algorithm to random feature represented functions (Rahimi and Recht, 2008) and neural networks. For simplicity, we specify the algorithms with either kernel, random feature representation or neural networks for both primal and dual functions. It should be emphasized that in fact the parametrization choice of the dual function is *independent* to the form of the primal function. Therefore, the algorithm can also be conducted in *hybrid setting* where the primal function uses one form of function approximator, while the dual function use another form of function approximator.

Instead of solving (10), in this section, we consider the alternative reformulation by penalizing the norm of the dual function, which has been widely used as an alternative to the constrained problem in machine learning literatures, and is proven to be more robust often times in practice,

$$\min_{f\in\mathcal{F}}\max_{u\in\mathcal{H}}\ \Phi(f,u)+\frac{\lambda_1}{2}\|f\|_{\mathcal{F}}^2-\frac{\lambda_2}{2}\|u\|_{\mathcal{H}}^2 \tag{24}$$

It is well-known that there is a one-to-one relation between $\delta_{\mathcal{F}}$, δ and λ_1, λ_2, respectively, such that the optimal solutions to (10) and (24) are the same. The objective can also be regarded as a smoothed approximation to the original problem of our interest, see (Nesterov, 2005). Problem (24) can be solved efficiently via our Algorithm 1 by simply revoking the projection operators.

E.1 Dual Random Feature Embeddings

In this section, we specify the proposed algorithm leveraging random feature to approximate kernel function. For arbitrary positive definite kernel, $k(x,x)$, there exists a measure $\mathbb{P}$ on $\mathcal{X}$, such that $k(x,x')=\int\widehat{\phi}_w(x)\widehat{\phi}_w(x')d\mathbb{P}(w)$ (Devinatz, 1953; Hein and Bousquet, 2004), where random feature $\widehat{\phi}_w(x):\mathcal{X}\to\mathbb{R}$ from $L_2(\mathcal{X},\mathbb{P})$. Therefore, we can approximate the function $f\in\mathcal{H}$ with Monte-Carlo approximation $\hat{f}\in\widehat{\mathcal{H}}^m=\{\sum_{i=1}^m\beta_i\widehat{\phi}_{\omega_i}(\cdot)|\|\beta\|_2\leqslant C\}$ where $\{w_i\}_{i=1}^m$ sampled from $\mathbb{P}(\omega)$ (Rahimi and Recht, 2009). With such approximation, we obtain the corresponding *dual random feature embeddings* variants.

Denote the random feature for $\tilde{k}(\cdot,\cdot)$ and $k(\cdot,\cdot)$ as $\widehat{\psi}_w(\cdot)$ and $\widehat{\phi}_w(\cdot)$ with respect to distribution $\widetilde{\mathbb{P}}(\omega)$ and $\mathbb{P}(\omega)$, respectively, we approximate the $f(\cdot)$ and $u(\cdot)$ by $\hat{f}(\cdot)=\theta^\top\widehat{\psi}(\cdot)$ and $\hat{u}(\cdot)=\eta^\top\widehat{\phi}(\cdot)$, where $\theta\in\mathbb{R}^{m\times1}$, $\eta\in\mathbb{R}^{p\times1}$, $\widehat{\psi}(\cdot)=[\widehat{\psi}_{\tilde{w}_1}(\cdot),\widehat{\psi}_{\tilde{w}_2}(\cdot),\ldots,\widehat{\psi}_{\tilde{w}_m}(\cdot)]^\top$ with $\{\tilde{w}_i\}_{i=1}^m\sim\widetilde{\mathbb{P}}(\omega)$ and $\widehat{\phi}(\cdot)=[\widehat{\phi}_{w_1}(\cdot),\widehat{\phi}_{w_2}(\cdot),\ldots,\widehat{\phi}_{w_m}(\cdot)]^\top$ with $\{w_i\}_{i=1}^p\sim\mathbb{P}(\omega)$. Then, we have the saddle point reformulation of (1),

$$\min_\theta\max_\eta\widehat{\Phi}(\theta,\eta):=\mathbb{E}_{x,y}\mathbb{E}_{z|x}\left[\theta^\top\widehat{\psi}(z,x)\widehat{\phi}(x,y)^\top\eta-l_y^*(\eta^\top\widehat{\phi}(x,y))\right]+\frac{\lambda_1}{2}\|\theta\|^2-\frac{\lambda_2}{2}\|\eta\|^2. \tag{25}$$

Apply the proposed algorithm to (25), we obtain the update rule in i-th iteration,

$$\begin{aligned}\theta_{i+1}&=&(1-\gamma_i\lambda_1)\theta_i-\gamma_i\hat{u}(x_i,y_i)\widehat{\psi}(z_i,x_i),\\ \eta_{i+1}&=&(1-\gamma_i\lambda_1)\eta_i+\gamma_i\big[\hat{f}_i(z_i,x_i)-\nabla\ell_{y_i}^*(\hat{u}(x_i,y_i))\big]\widehat{\phi}(x_i,y_i).\end{aligned}$$

We emphasize that with the random feature representation will introduce an extra approximation error term in the order of $\mathcal{O}(1/\sqrt{m})$. To balance the approximate error and the statistical generalization error, we must use m sufficiently large.

E.2 Extension to Embedding Doubly-SGD

To alleviate the approximation error introduced by random feature representation, we can further generalize the algorithmic technique about doubly stochastic gradient (Dai et al., 2014) to the saddle point problem (24), which can be viewed as setting m to be infinite conceptually, therefore, eliminate the approximation error due to random feature representation. The embedding doubly-SGD is illustrated in Algorithm 2,

E.3 Dual Neural Networks Embeddings

To achieve better performance with fewer basis functions, we can also learn the basis functions $\widehat{\psi}(\cdot)$ and $\widehat{\phi}(\cdot)$ jointly with θ and η by back-propagation. Specifically, denote the parameters in $\widehat{\psi}(\cdot)=[\widehat{\psi}_{\tilde{w}_1}(\cdot),\widehat{\psi}_{\tilde{w}_2}(\cdot),\ldots,\widehat{\psi}_{\tilde{w}_m}(\cdot)]^\top$ and $\widehat{\phi}(\cdot)=[\widehat{\phi}_{w_1}(\cdot),\widehat{\phi}_{w_2}(\cdot),\ldots,\widehat{\phi}_{w_m}(\cdot)]^\top$ explicitly as $\widetilde{W}=[\tilde{w}_i]_{i=1}^m$ and $W=[w_i]_{i=1}^m$, we also include $\tilde{W}$ and W into optimization (25), which results

$$\min_{\theta,\widetilde{W}}\max_{\eta,W}\widehat{\Phi}(\theta,\widetilde{W},\eta,W):=\mathbb{E}_{x,y}\mathbb{E}_{z|x}\left[\theta^\top\widehat{\psi}_{\widetilde{W}}(z,x)\widehat{\phi}_W(x,y)^\top\eta-l_y^*\left(\eta^\top\widehat{\phi}_W(x,y)\right)\right]+\frac{\lambda_1}{2}\|\theta\|^2-\frac{\lambda_2}{2}\|\eta\|^2. \tag{26}$$

Algorithm 2 Embedding-Doubly SGD for (24)

```
Input: $\mathbb{P}(x, y)$, $\mathbb{P}(z|x)$, $\mathbb{P}(\omega)$, $\{\gamma_i \geqslant 0\}_{i=1}^t$
for $i = 1, \ldots, t$ do
    Sample $x_i, y_i \sim \mathbb{P}(x, y)$.
    Sample $z_i \sim \mathbb{P}(z|x_i)$.
    Sample $\omega_i \sim \mathbb{P}(\omega)$ with seed $i$
    Sample $\widetilde{\omega}_i \sim \widetilde{\mathbb{P}}(\omega)$ with seed $\tilde{i}$
    Compute $f_i =$ Predict$(z_i, x_i, \{\alpha_j\}_{j=1}^i)$
    Compute $u_i =$ Predict$(x_i, y_i, \{\beta_j\}_{j=1}^i)$
    $\alpha_{i+1} = \gamma_i u_i(x_i, y_i)\widehat{\psi}_{\widetilde{\omega}_i}(z_i, x_i)$.
    $\beta_{i+1} = \gamma_i[f_i(z_i, x_i) - \nabla \ell_{y_i}^*(u_i(x_i, y_i))]\widehat{\phi}_{\omega_i}(x_i, y_i)$.
    for $j = 1, \ldots, i$
    $\alpha_j = (1 - \gamma_i\lambda_1)\alpha_j$, $\beta_j = (1 - \gamma_i\lambda_2)\beta_j$
end for
```

Algorithm 3 $u =$ **Predict**$(x, y, \{\beta_i\}_{i=1}^t)$

```
Require: $\mathbb{P}(\omega)$, $\widehat{\phi}_\omega(x, y)$.
Set $u = 0$.
for $i = 1, \ldots, t$ do
    Sample $\omega_i \sim \mathbb{P}(\omega)$ with seed $i$.
    $u = u + \beta_i \widehat{\phi}_{\omega_i}(x, y)$.
end for
```

Algorithm 4 $f =$ **Predict**$(z, x, \{\alpha_i\}_{i=1}^t)$

```
Require: $\widetilde{\mathbb{P}}(\omega)$, $\widehat{\psi}_\omega(z, x)$.
Set $f = 0$.
for $i = 1, \ldots, t$ do
    Sample $\widetilde{\omega}_i \sim \widetilde{\mathbb{P}}(\omega)$ with seed $\tilde{i}$.
    $f = f + \alpha_i \widehat{\psi}_{\widetilde{\omega}_i}(z, , x)$.
end for
```

Apply the proposed algorithm to (26), we obtain the update rule for all the parameters, $\{\theta, \eta, \widetilde{W}, W\}$, in i-th iteration,

$$\begin{aligned}
\theta_{i+1} &= (1 - \gamma_i\lambda_1)\theta_i - \gamma_i \eta_i^\top \widehat{\phi}_{W_i}(x_i, y_i)\widehat{\psi}_{\widetilde{W}_i}(z_i, x_i), \\
\eta_{i+1} &= (1 - \gamma_i\lambda_1)\eta_i + \gamma_i \left[\theta_i^\top \widehat{\psi}_{\widetilde{W}_i}(z_i, x_i) - \nabla \ell_{y_i}^* \left(\eta_i^\top \widehat{\phi}_{W_i}(x_i, y_i)\right)\right] \widehat{\phi}_{W_i}(x_i, y_i), \\
\widetilde{W}_{i+1} &= \widetilde{W}_i - \gamma_i \eta_i^\top \widehat{\phi}_{W_i}(x_i, y_i)\theta_i^\top \nabla_{\widetilde{W}} \widehat{\psi}_{\widetilde{W}}(z_i, x_i), \\
W_{i+1} &= W_i + \gamma_i \left[\theta_i^\top \widehat{\psi}_{\widetilde{W}_i}(z_i, x_i) - \nabla \ell_{y_i}^* \left(\eta_i^\top \widehat{\phi}_{W_i}(x_i, y_i)\right)\right] \eta_i^\top \nabla_W \widehat{\phi}_W(x_i, y_i).
\end{aligned}$$

Here we only demonstrate the back-propagation algorithm applies to one-layer basis functions, in fact, it can be extended to the deep basis functions, *i.e.*, hierarchical composition functions, straight-forwardly if necessary. With such deep neural networks as function approximator in our algorithm, we achieve the dual neural networks embeddings.

F Experimental Details

In all experiemnts, we conduct comparison on algorithms to optimize the objective with regularization on both primal and dual functions. Since the target is evaluating the performance of algorithms on the same problem, we fix the weights of the regularization term for the proposed algorithm and the competitors for fairness. The other paramters of models and algorithms, *e.g.*, step size, mini-batch size, kernel parameters and so on, are set according to different tasks.

F.1 Learning with Invariance

Noisy in measurements. We select the best $\eta \in \{0.1, 1, 10\}$ and $n_0 \in \{1, 10, 100\}$. We use Gaussian kernel for both primal and dual function, whose bandwidth σ are selected from $\{0.05, 0.1, 0.15, 0.2\}$. We set the batch size to be 50. In testing phase, the observation is noisyless.

QuantumMachine. We selected the stepsize parameters $\eta \in \{0.1, 0.5, 1\}$ and $n_0 \in \{100, 1000\}$. We adopted Gaussian kernel whose bandwidth is selected by median trick with coeffcient in $\{0.1, 0.25, 0.5, 1\}$. The batch size is set to be 1000. To illustrate the benefits of sample efficiency, we generated 10 virtual samples in training phase and 20 in testing phase.

F.2 Policy Evaluation

We evaluated all the algorithms in terms of mean square Bellman error on the testing states. On each state s, the mean square Bellman error is estimated with 100 next states s' samples. We set the number of the basis functions in GTD2 and RG to be 2^8. To achieve the convergence property based the theorem (Nemirovski et al., 2009), we set stepsize to be be $\frac{\eta}{n_0+\sqrt{t}}$ in the proposed algorithm and GTD2, $\frac{\eta}{n_0+t}$ in Kernel MDP and RG.

Navigation. The batch size is set to be 20. $\{\eta, n_0\} \in \{0.1, 1, 10\}$. We adopted Gaussian kernel and select the best primal and dual kernel bandwidth in range $\{0.01, 0.05, 0.1, 0.15, 0.2\}$. The γ in MDP is set to be 0.9.

Cart-pole swing up. The batch size is set to be 20. The stepsize parameters are chosen in range $\{\eta, n_0\} \in \{0.05, 0.2, 1, 10, 50, 100\}$. We adopted Gaussian kernel and the primal and dual kernel bandwidth are se lected by median trick with coeffcient in $\{0.5, 1, 5, 10\}$. The γ is set to be 0.96. The reward is $R(s) = \frac{1}{2}(s_1^2 + s_2^2 + s_3^2 + 5(s_4 - \pi)^2)$ where the states s_1, s_2, s_3, s_4 are the cart position, cart velocity, pendulum velocity and pendulum angular position.

PUMA-560 manipulation. The batch size is set to be 20. Step-size parameters are chosen in range $\{\eta, n_0\} \in \{0.05, 0.1, 5, 10, 100, 500\}$. We adopted Gaussian kernel and the primal and dual kernel bandwidth are selected by median trick with coeffcient in $\{0.5, 1, 5, 10\}$. The γ is set to be 0.9. The reward is $R(s) = \frac{1}{2}(\sum_{i=1}^{4}(s_i - \frac{\pi}{4})^2 + \sum_{i=5}^{6}(s_i + \frac{\pi}{4})^2 + \sum_{i=7}^{12} s_i^2)$ where $s_1, \ldots, s_6$ and $s_7, \ldots, s_{12}$ are joint angles and velocities, respectively.