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

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

where $D^2_{\mathcal{F}} = \sup_{f\in\mathcal{F}} \frac{1}{2}\|f_1 - f\|_2^2$ and $D^2_{\mathcal{H}} = \sup_{u\in\mathcal{H}^\delta} \frac{1}{2}\|u_1 - u\|^2_{\mathcal{H}} \leqslant 2\delta$. 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',

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

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

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

Plugging in $M_1^2 = 2\kappa(M_{\mathcal{F}} + c_\ell)$ and $M_2^2 = \frac{1}{4}(\delta+\kappa)^2 C_{\mathcal{F}}$ 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

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

Minimizing over $\gamma > 0$, we get the "theoretical" optimal choice for the γ as $\gamma^* = \sqrt{\frac{2D^2_{\mathcal{F}} + 4\delta}{\mathcal{C}(\delta,\kappa)}}$. 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

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

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

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

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,

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

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 $f_i = \textbf{Predict}(z_i, x_i, \{\alpha_j\}_{j=1}^i)$
    Compute $u_i = \textbf{Predict}(x_i, y_i, \{\beta_j\}_{j=1}^i)$
    $\alpha_{i+1} = \gamma_i u_i(x_i, y_i)\widehat{\psi}_{\widetilde{\omega}_i}(z_i, x_i)$.
    $\beta_{i+1} = \gamma_i[f_i(z_i, x_i) - \nabla \ell_{y_i}^*(u_i(x_i, y_i))]\widehat{\phi}_{\omega_i}(x_i, y_i)$.
    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.