Is Model Ensemble Necessary? Model-Based RL Via A Single Model With Lipschitz Regularized Value Function

BY ZHIJUN ZENG

2023年5月25日

目录

1 Preliminaries And Background

MDP: $(\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{P}_0, r, \gamma)$, $\mathcal{P}(s'|s,a)$ transition, $\mathcal{P}_0(s)$ initial distribution, $r(s,a)$ reward function.

Optimal Bellman Operator: $\mathcal{T}^*$: $\mathcal{T}^*Q(s,a) = r(s,a) + \gamma \int \mathcal{P}(s'|s,a) \mathrm{argmax}_{a'} Q(s', a') \mathrm{ds}'$. The goal of value based RL is to learn Q_ϕ to approximate the optimal state-action value function Q^*, where $\mathcal{T}^*Q^* = Q^*$.

For model-based RL , we denote the approximate model for state transition and reward function as $\hat{\mathcal{P}}_\theta$ and $\hat{r}_\theta$. Similarly, we define the model-induced Bellman Operator $\hat{\mathcal{T}}^*$ such that

$$\hat{\mathcal{T}}^*Q(s,a) = \hat{r}(s,a) + \gamma \int \hat{\mathcal{P}}(s'|s,a) \mathrm{argmax}_{a'} Q(s', a') \mathrm{ds}'$$

1.1 Probabilistic Dynamics Model Ensemble

Probabilistic dynamics model ensemble consists of K neural networks $\hat{\mathcal{P}}_\theta = \{\hat{\mathcal{P}}_{\theta_1}, \ldots, \hat{\mathcal{P}}_{\theta_K}\}$ with the same architecture but random initialized with different weight.

Given (s, a) ,the prediction of each neural network is a Gaussian distribution with diagonal covariances of the next state:

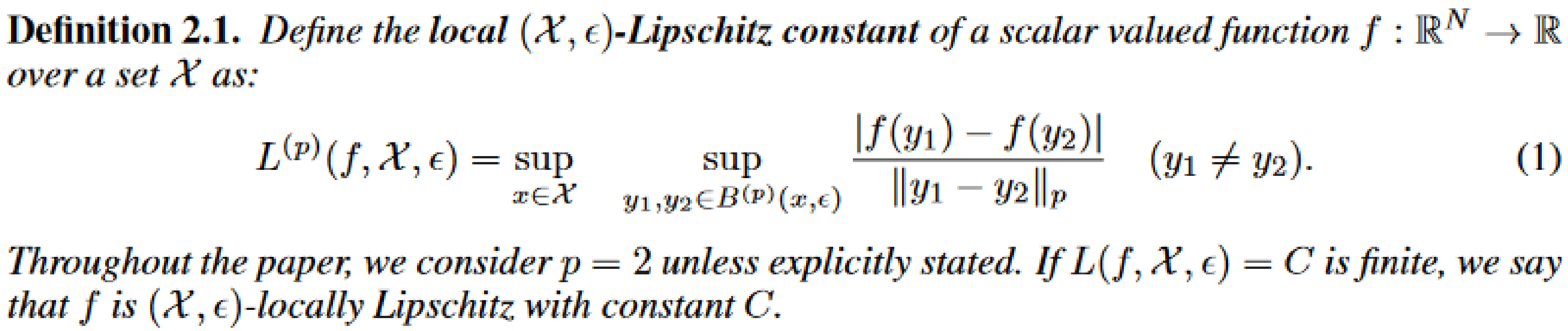
$$\hat{\mathcal{P}}_{\theta_k}(s'|s, a) = \mathcal{N}(\mu_{\theta_k}(s, a), \Sigma_{\theta_k}(s, a))$$

The model is trained using negative log likelyhood loss

$$\mathcal{L}(\theta_k) = \sum_{t=1}^{N} [\mu_{\theta_k}(s_t, a_t) - s_{t+1}]^T \Sigma_{\theta_k}(s_t, a_t)^{-1} [\mu_{\theta_k}(s_t, a_t) - s_{t+1}] + \operatorname{logdet} \Sigma_{\theta_k}(s_t, a_t)$$

During model rollouts, the probabilistic dynamics model ensemble first randomly selects a network from the ensemble and then samples the next state from the predicted Gaussian distribution.

1.2 Local Lipschitz Constant



Definition 2.1. *Define the local $(\mathcal{X}, \epsilon)$-Lipschitz constant of a scalar valued function $f : \mathbb{R}^N \to \mathbb{R}$ over a set $\mathcal{X}$ as:*

$$L^{(p)}(f, \mathcal{X}, \epsilon) = \sup_{x \in \mathcal{X}} \sup_{y_1, y_2 \in B^{(p)}(x, \epsilon)} \frac{|f(y_1) - f(y_2)|}{\|y_1 - y_2\|_p} \quad (y_1 \neq y_2). \tag{1}$$

Throughout the paper, we consider $p = 2$ unless explicitly stated. If $L(f, \mathcal{X}, \epsilon) = C$ is finite, we say that f is $(\mathcal{X}, \epsilon)$-locally Lipschitz with constant C.

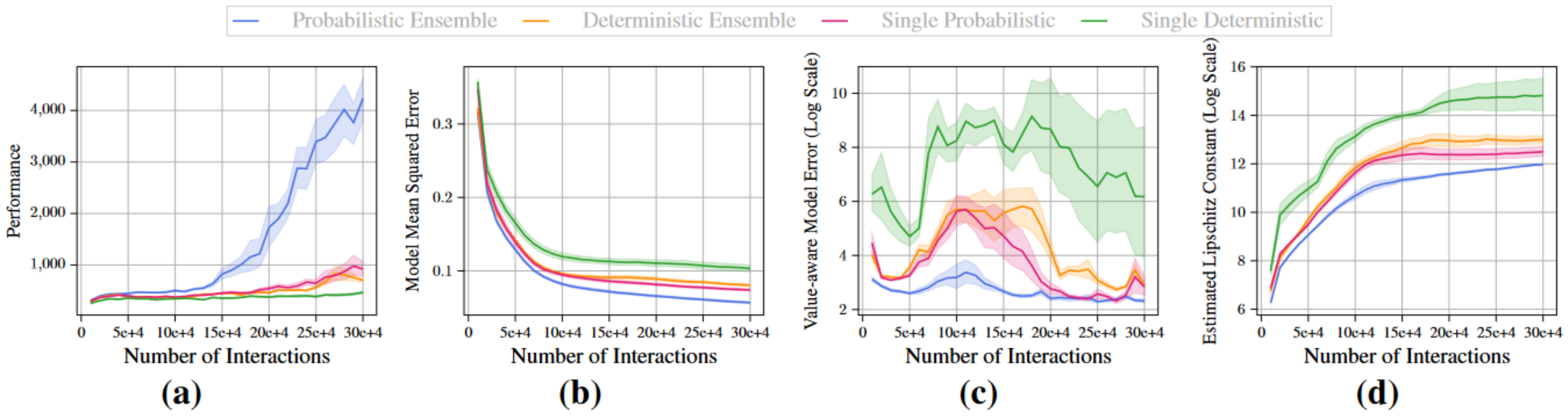
Global Lipschitz constant is equivalent to $L(f, \mathbb{R}^N, \varepsilon)$.

2 Local Lipschitz Condition Of Value Functions In MBRL

2.1 Implicit Regularization of Value Network By Probablistic Ensemble Model

The experiment is Humanoid with MBPO and SAC as the backbone algorithm for policy and value optimization.

We run MBPO with four different types of trained environment models: (1) a single deterministic dynamics model, (2) a single probabilistic dynamics model, (3) an ensemble of seven deterministic dynamics models, and (4) an ensemble of seven probabilistic dynamics models.



Algorithm 2 Model-Based Policy Optimization with Deep Reinforcement Learning

```
1:  Initialize policy $\pi_\phi$, predictive model $p_\theta$, environment dataset $\mathcal{D}_{\text{env}}$, model dataset $\mathcal{D}_{\text{model}}$
2:  for $N$ epochs do
3:      Train model $p_\theta$ on $\mathcal{D}_{\text{env}}$ via maximum likelihood
4:      for $E$ steps do
5:          Take action in environment according to $\pi_\phi$; add to $\mathcal{D}_{\text{env}}$
6:          for $M$ model rollouts do
7:              Sample $s_t$ uniformly from $\mathcal{D}_{\text{env}}$
8:              Perform $k$-step model rollout starting from $s_t$ using policy $\pi_\phi$; add to $\mathcal{D}_{\text{model}}$
9:          for $G$ gradient updates do
10:             Update policy parameters on model data: $\phi \leftarrow \phi - \lambda_\pi \hat{\nabla}_\phi J_\pi(\phi, \mathcal{D}_{\text{model}})$
```

2.1.1 Value-aware model error

In MBPO algorithm, it fits an ensemble of M Gaussian environment models

$$\widehat{\mathcal{P}_\theta^{(i)}}(\cdot|s,a)=\mathcal{N}(\mu_\theta^{(i)}(s,a),\Sigma_{\theta_k}^{(i)}(s,a)),i=1,\ldots M$$

The mean squared error $\mathbb{E}_{(s,a)\sim\rho,s'\sim\mathcal{P}(s'|s,a),\hat{s}'\sim\widehat{\mathcal{P}}_\theta(\cdot|s,a)}[\|s'-\hat{s}'\|^2]$ is often small.Here is another measurement of quality of learned transition model:

$$\begin{aligned}\mathcal{L}_{\text{vame}}(\hat{\mathcal{P}}_\theta;Q,\rho) &= \mathbb{E}_{(s,a)\sim\rho}[(\mathcal{T}^*Q(s,a)-\hat{\mathcal{T}}^*Q(s,a))^2]\\ &= \gamma^2\ \mathbb{E}_{(s,a)\sim\rho}\left[\left(\int\mathcal{P}(s'|\ s,\ a)\ V(s')\ d\ s' - \int\hat{\mathcal{P}}_\theta(\hat{s}'|\ s,\right.\right.\\ &\left.\left. a)\,V(\hat{s}')\,d\,\hat{s}'\right)^2\right]\\ &\leq \gamma^2\ \mathbb{E}_{(s,a)\sim\rho,s'\sim\mathcal{P}(\cdot|s,a),\hat{s}'\sim\hat{\mathcal{P}}_\theta(\cdot|s,a)}[(V(\hat{s}')-V(s'))^2],\ V(s)=\\ &\max_a Q(s,a)\end{aligned}$$

Even though the learned transition model is approximately accurate, the model prediction error can still be amplified by the value function so that the two Bellman operators yield completely different updates on the value function.

2.1.2 Value function Lipschitz regulates value-aware model error

Although the probabilistic ensemble model and single deterministic model achieve similar mean squared errors, value functions trained from an ensemble of probabilistic dynamics model has a significantly smaller Lipschitz constant and value-aware model error.

We view the Gaussian transition models $\hat{\mathcal{P}}_\theta^{(i)} = f_\theta^{(i)} + g_\theta^{(i)}$ where $f_\theta^{(i)}$ is deterministic and $g_\theta^{(i)}(\cdot|s, a) = \mathcal{N}(0, \Sigma_\theta^{(i)}(s, a))$ is noise distribution with zero mean.The training target value is

$$r(s,a) + \gamma \mathbb{E}_{i \sim \mathrm{Cat}\left(M, \frac{1}{M}\right), \varepsilon_i \sim g_\theta^{(i)}(\cdot|s,a)}[V(f_\theta^{(i)}(s,a) + \varepsilon_i)]$$

Here we view data generated from the probabilistic ensemble model as an implicit augmentation.

The augmentation comes from two sources:

(i) variation of prediction across different models in the ensemble

(ii) the noise added by each noise distribution $g_\theta^{(i)}$.

By augmenting the transition with a diverse set of noise and then training the value functions with such augmented samples, it implicitly regularizes the local Lipschitz condition of the value network over the local region around the state where the model prediction is uncertain.

2.2 Error Bound Of Model-Based Value Iteration With Locally Lipschitz Value Functions

Target: Analyze how the Lipschitz constant of the value function affects the learning dynamics of the model-based (approximate) value iteration algorithm.

Assumption: we assume that we have the true reward function r(s, a).

Value iteration Algorithm: at k-iteration, we obtained dataset $\mathcal{D}_k = \{(s_i, a_i, s_i')\}_{i=1}^N$, where (s_i, a_i) sampled iid from $\rho \in \Delta(\mathcal{S} \times \mathcal{A})$, the empirical state-action distribution, $s_i' \sim \mathcal{P}(\cdot|s_i, a_i)$ is under environment transition.We approximate the transition kernel $\hat{\mathcal{P}}$ by minimizing

$$\hat{\mathcal{P}}_k \leftarrow \arg\min_{\hat{\mathcal{P}} \in \mathcal{M}} \frac{1}{N} \sum_{i=1}^{N} \int \hat{\mathcal{P}}(\hat{s}'|\, s_i, a_i)\, |\hat{s}' - s_i'|\, d\, \hat{s}'$$

The k-th iteration of value update is

$$\hat{Q}_k \leftarrow \arg\min_{\hat{Q}\in\mathcal{F}} \mathcal{L}_{reg}(\hat{Q}; \hat{Q}_{k-1}, \hat{\mathcal{P}}_k) := \arg\min_{\hat{Q}\in\mathcal{F}} \mathbb{E}_{(s,a)\sim\rho}[(\hat{Q}(s,a) - \hat{\mathcal{T}}^* \hat{Q}_{k-1}(s,a))^2]$$

empirically we use the follow formula

$$\hat{Q}_k \leftarrow \arg\min_{\hat{Q}\in\mathcal{F}} \frac{1}{N} \sum_{i=1}^{N} \left| \hat{Q}(s_i, a_i) - \left(r_i + \gamma \int \hat{\mathcal{P}}_k(\hat{s}' | s_i, a_i) \hat{V}_{k-1}(\hat{s}') \, d\hat{s}' \right) \right|^2$$

where $\hat{V}_{k-1}(\hat{s}') = \max_{a'} \hat{Q}_{k-1}(\hat{s}', a')$.

To simplify the analysis, we assume that we are given a fixed state-action distribution ρ such that for every iteration, we can sample i.i.d. from this data distribution.

Algorithm 1 Model-based Approximate Value Iteration on Inverted Pendulum

```
$K$ : Total number of iterations for the algorithm
$H$ : Number of gradient steps to solve the inner optimization
$\mathcal{M}, \mathcal{G}$ : Space of transition probability kernels and reward functions
$\mathcal{F}$: Space of value function
$\mu \in \Delta(\mathcal{S} \times \mathcal{A})$: a data-collecting state-action distribution
Sample i.i.d from $\mu$ to generate the training dataset $\mathcal{D} = \{(s_i, a_i, s'_i, r_i)\}_{i=1}^{N}$
$\hat{\mathcal{P}} \leftarrow \arg\min_{\hat{\mathcal{P}} \in \mathcal{M}} \sum_{i=1}^{N} \|s'_i - \int \hat{\mathcal{P}}(ds'|s,a)s'\|^2$
$\hat{r} \leftarrow \arg\min_{\hat{r} \in \mathcal{G}} \sum_{i=1}^{N} (r_i - \hat{r}(s_i, a_i))^2$
Initialize the value function $\hat{Q}_0$.
**repeat**
  **for** $k = 0$ **to** $K - 1$ **do**
    Sample i.i.d $N$ state-action pairs from $\rho$ : $\{(s_i, a_i)\}_{i=1}^{N}$
    Compute $\hat{s}'_i \sim \mathcal{P}(\cdot|s_i, a_i)$, $\hat{r}_i = \hat{r}(s_i, a_i)$
    **for** $t = 0$ **to** $H - 1$ **do**
      Update policy using gradient ascent with $\frac{1}{N} \sum_{i=1}^{N} \nabla_\theta \hat{Q}_\phi(s_i, \pi_\theta(s_i))$
    **end for**
    **for** $t = 0$ **to** $H - 1$ **do**
      Update value function using gradient descent with
      $\frac{1}{N} \sum_{i=1}^{N} \nabla_\phi \Big(\hat{r}_i + \gamma \hat{Q}_\phi(\hat{s}'_i, \pi_\theta(\hat{s}'_i)) - \hat{Q}_\phi(s_i, a_i)\Big)^2$
    **end for**
  **end for**
**until** end of training
**Output:** $\hat{Q}_\phi$
```

2.2.1 Assumption

Assumption 3.1. *The environment transition is deterministic.*

Next, to apply the concentration inequality in our analysis, we have to make the following technical assumption of the state space and reward function.

Assumption 3.2. ***(Boundedness of State Space and Reward Function)*** *There exists constants D, $R_{\max}$ such that for all $s \in \mathcal{S}, a \in \mathcal{A}$, $\|s\|_2 \leq D$ and $r(s,a) < R_{\max}$.*

Assumption 3.3. ((ϵ, ρ)***-Approximate Realizability***)

$$\inf_{\hat{\mathcal{P}} \in \mathcal{M}} \mathcal{L}_2(\hat{\mathcal{P}}) := \inf_{\hat{\mathcal{P}} \in \mathcal{M}} \int \hat{\mathcal{P}}(d\hat{s}'|s,a) \big\| \mathcal{P}(s,a) - \hat{s}' \big\|_2 d\rho(s,a) \leq \epsilon \tag{5}$$

We also make a critical assumption of the local Lipschitz condition of the value function class. In particular, for the state-action value function $Q : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$, we define that Q is (ϵ, p)-locally Lipschitz with constant L if for every $a \in \mathcal{A}$, the function $Q_a : s \mapsto Q(s,a)$ is (ϵ, p)-locally Lipschitz with constant less than or equal to L.

Assumption 3.4. ***(Local Lipschitz condition of value functions)*** *There exists a finite L, such that for every $Q \in \mathcal{F}$, it is $(\mathcal{X}, 2\epsilon)$-locally Lipschitz with a constant less than or equal to L, where $\mathcal{X}$ is the support of the distribution ρ.*

Theorem 3.6. *Suppose $\hat{Q}_0$ is initialized such that $\hat{Q}_0(s,a) \leq \frac{R_{\max}}{1-\gamma}$ for $\forall\ (s,a)$. Under the assumptions of 3.1, 3.2, 3.3, 3.4, and 3.5, after K iterations of the model-based approximate value iteration algorithm, there exists a constant $\kappa(\alpha)$ which depends solely on $\alpha \in (0,1)$ such that,*

$$\mathbb{E}_{(s,a)\sim\mathcal{P}_0}\left[\left|Q^*(s,a) - \hat{Q}_K(s,a)\right|\right] \leq \frac{2\gamma}{(1-\gamma)^2}\Big[C(\rho,\mathcal{P}_0)\Big(\max_{0\leq k\leq K}\delta_k + \gamma^2(4\epsilon^2L^2\xi + \frac{(1-\xi)R_{\max}^2}{(1-\gamma)^2})\Big) + 2\gamma^K R_{\max}\Big] \tag{7}$$

where $\delta_k^2 = \mathcal{L}_{reg}(\hat{Q}_k; \hat{Q}_{k-1}, \hat{\mathcal{P}}_k)$ is the regression error defined in Equation (3), $\xi = 1 - \exp(-\frac{\epsilon N^{\frac{1}{1+\alpha}}}{\kappa(\alpha)D^2R^{\frac{2\alpha}{1+\alpha}}})$, and $C(\rho,\mathcal{P}_0)$ is the concentrability constant defined in Definition A.1.

注意 1. As $N \to \infty$,$\xi \to 1$, the value-aware model error $\mathbb{E}_{(s,a)\sim\rho}[(\mathcal{T}^* Q(s,a) - \hat{\mathcal{T}}^* Q(s,a))^2]$ will be govern by $4\varepsilon^2 L^2 \xi$ which is controlled by the local lipschtiz constant L.

注意 2. As $k \to \infty$, LHS is bounded by $\frac{2\gamma C(\rho, P_0)}{(1-\gamma)^2}(\max_{0 \leqslant k \leqslant K} \delta_k + 4\varepsilon^2 L^2 \xi)$. The local Lipschitz constant has played in controlling the suboptimality of the algorithm.

注意 3. If the learned transition model has a low prediction error $L_2(\hat{\mathcal{P}})$ the local Lipschitz constant of the value function is bounded, then we can still have a small value-aware model error and get a similar error propagation result.

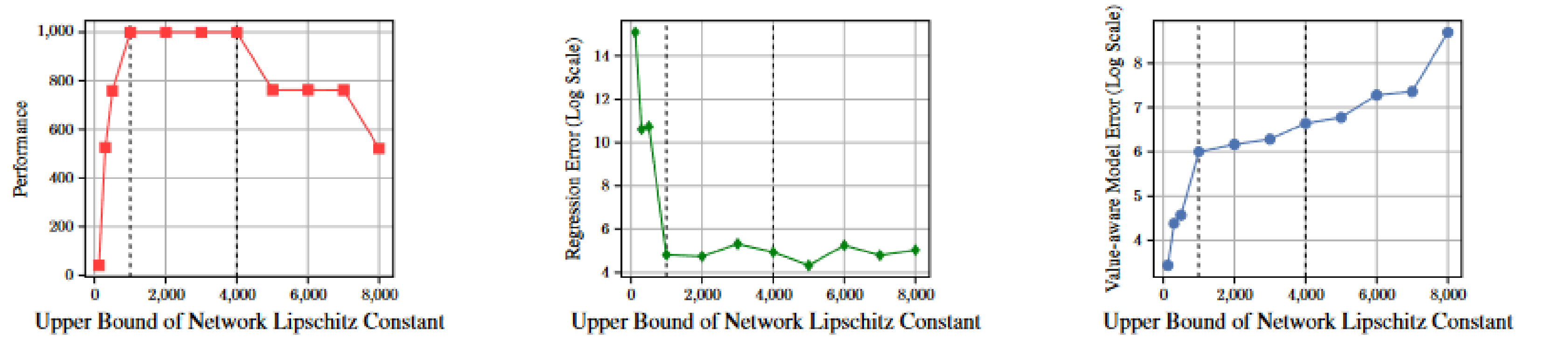
2.2.2 Trade of between regression error and value aware model error

As L gets smaller, the model-induced Bellman operator gets closer to the actual underlying Bellman operator. But we also impose a stronger condition on the value function class , thus the value function space shrink, δ_k is expected to get larger

$$\hat{Q}_k \leftarrow \arg\min_{\hat{Q}\in\mathcal{F}} \mathcal{L}_{reg}(\hat{Q}; \hat{Q}_{k-1}, \hat{\mathcal{P}}_k) := \arg\min_{\hat{Q}\in\mathcal{F}} \mathbb{E}_{(s,a)\sim\rho}[(\hat{Q}(s,a) - \hat{\mathcal{T}}^* \hat{Q}_{k-1}(s,a))^2]$$

Experiment:Inverted Pendulum

Run model-based approximate value iteration algorithms for 1000 iterations.



(a) best evaluation performance (b) max regression error (c) max value-aware model error

Figure 2: Model-based value iteration on Inverted-Pendulum across value networks with different Lipschitz constraints. All the results are the median over 5 random seeds.

3 Method

3.1 Spectral Normalization

By controlling the spectral norm of the weight matrix at every layer of the value network, its Lipschitz constant is upper bounded.

In particular, at every forward pass, we approximate the spectral radius of the weight matrix with one step of power iteration.

$$\begin{aligned} \boldsymbol{v} \leftarrow W\boldsymbol{u}^{(t-1)}; \alpha \leftarrow \|\boldsymbol{v}\|; \quad & \boldsymbol{v}^{(t)} \leftarrow \alpha^{-1}\boldsymbol{v} \\ \boldsymbol{u} \leftarrow W^T\boldsymbol{v}^{(t)}; \rho \leftarrow \|\boldsymbol{v}\|; \quad & \boldsymbol{u}^{(t)} \leftarrow \rho^{-1}\boldsymbol{u} \end{aligned}$$

Then $W := \max\left(1, \frac{\max(\alpha, \rho)}{\beta}^{-1} W\right)$. So the spectral norm will be clipped to β if it's bigger than λ, unchanged otherwise.

3.2 Robust Regularization

$$\mathcal{L}_{\text{reg}}^{(\alpha)}\left(Q_\phi; \epsilon, \pi_\psi, \{s_i, a_i, s_i'\}_{i=1}^{D}\right) = \sum_{i=1}^{D} \max_{|\tilde{s}_i - s_i|_\alpha \leq \epsilon} \left(Q_\phi(\tilde{s}_i, \pi_\psi(s_i')) - Q_\phi(s_i, a_i)\right)^2$$

This robust loss is to guarantee that the variation of the value function locally is small.Then we combine it with the original loss of the value function

$$\mathcal{L}(Q_\phi; \pi_\psi, \{s_i, a_i, s_i'\}_{i=1}^{D}) = \mathcal{L}_{\text{org}}\left(Q_\phi; \pi_\psi, \{s_i, a_i, s_i'\}_{i=1}^{D}\right) + \lambda\, \mathcal{L}_{\text{reg}}^{(\alpha)}\left(Q_\phi; \epsilon, \pi_\psi, \{s_i, a_i, s_i'\}_{i=1}^{D}\right)$$

The constrained optimization problem can be solved by Projected Gradient Descent and Fast gradient sign method.

4 Result

4.1 EMPIRICAL EVALUATIONS OF PROPOSED MECHANISMS

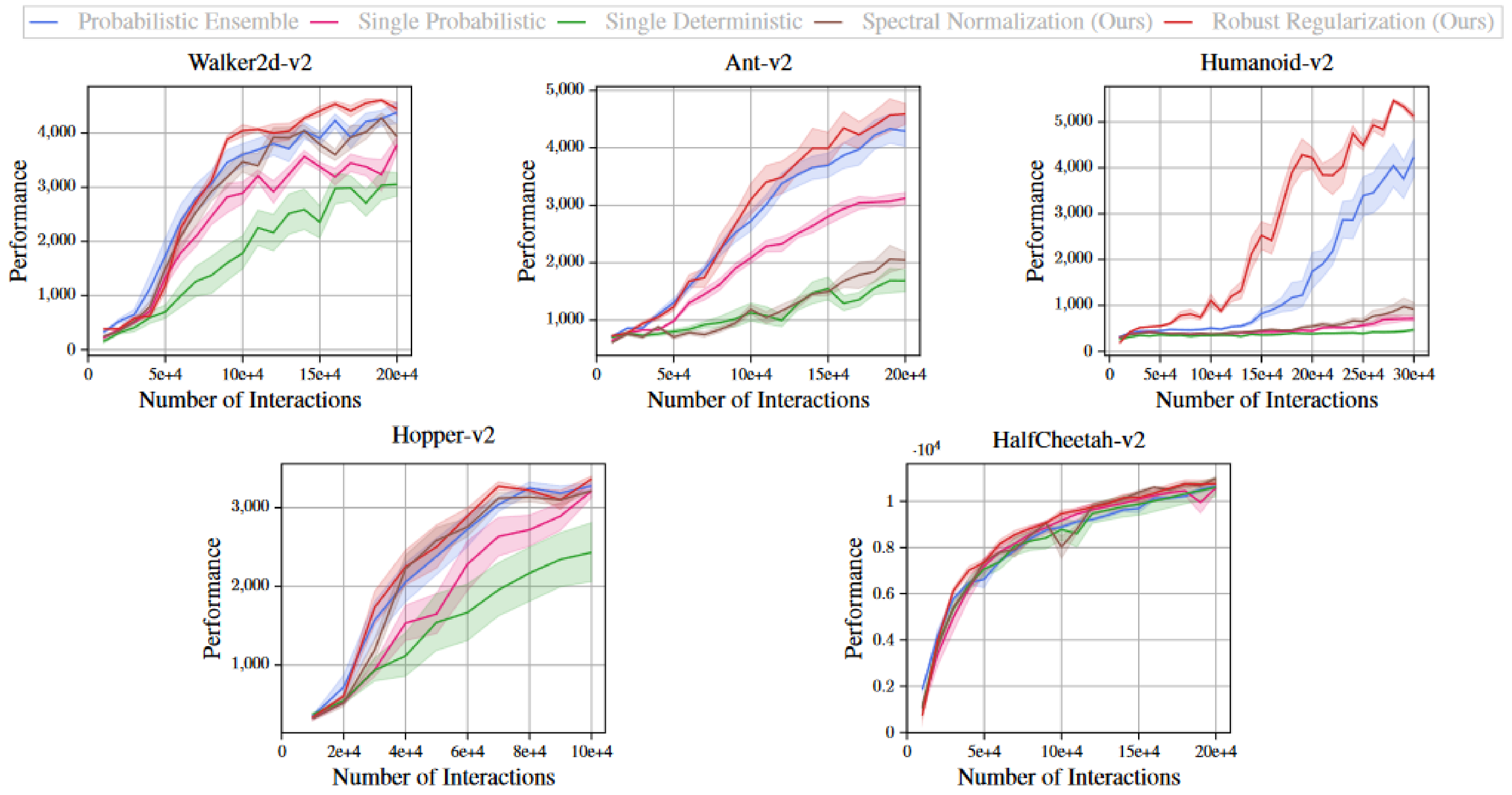
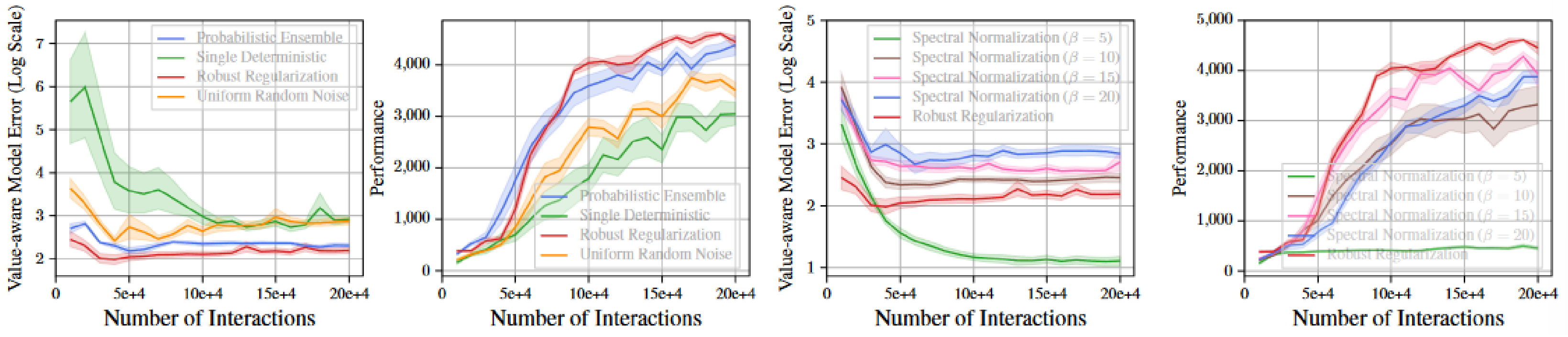


Figure 3: Performance of the proposed value function training mechanisms against baselines. Results are averaged over 8 random seeds and shaded regions correspond to the 95% confidence interval among seeds.

4.2 VISUALIZING THE VALUE-AWARE MODEL ERROR


Value-aware Model Error (Log Scale)
Performance
Number of Interactions
Probabilistic Ensemble
Single Deterministic
Robust Regularization
Uniform Random Noise
Spectral Normalization ($\beta = 5$)
Spectral Normalization ($\beta = 10$)
Spectral Normalization ($\beta = 15$)
Spectral Normalization ($\beta = 20$)
5e+4
10e+4
15e+4
20e+4