Learning Value Equivalent Models on Policy Improvement Path

1 Introduction

The basic intuition of value-aware model learning [6, 5] is to assign different weights to the transitions according to the *optimal* state value. However, the optimal value function is not known and the VAML loss cannot be readily minimized. Thus, as shown in [8, 7], we may find a model using the value equivalence principle, which is then used for value-based planning. As more value functions are augmented to be considered, the model will less compromise the performance, and eventually collapse to the real environment model.

Intuitively, the model that is value equivalent (VE) with $\{V\} = \{V^\pi | \pi \in \Pi\}$ accurately describe the real environment when Π contains all possible policies. However, in most cases we cannot enumerate all possible policies and their corresponding values. One potential solution is to train the VE model with *representative* policies (and their values). The key is to determine the principled set Π_t at iteration t in the training process that characterizes an effective VE model, both during training efficient and asymptotically optimal.

In this work, we focus on model-based RL in two frameworks and build the connections between the value equivalence principle and online model learning. Firstly, we study sample-based policy search with VE models, where the model generated samples are used to perform (model-free) policy optimization. We show that the convergence rate depends on the Rademacher complexity of the model loss, for which we design several online learners. In general, learning VE models through the policy improvement path gives a $T^{-\frac{1}{2}}$ rate of convergence for natural policy gradient. We also analyze the iterative model learning approaches in convex settings such as linear approximations. In nonconvex settings, we provide several extensions inspired by FTPL and optimistic FTPL with predictors for tighter regret bound.

We also analyze the wider usage of value equivalence principle in model-based planning. Modern agents such as MuZero have shown the superiority of VE models in complex tasks. Recent work [7] has observed even better experimental performance when constraining the model update w.r.t. $\Pi_t = \{\pi_i\}_{i=1}^{t-1}$, instead of $\Pi_t = \{\pi_{t-1}\}$ as used in MuZero. We investigate such phenomenon and provide theoretical guidance for designing the policy cover Π_t. Besides, from the local regret perspective, we prove an approximate local optimality with an online model learner.

2 Sample-Based Policy Search with VE Models

2.1 Convergence Rate of Natural Policy Gradient

In this section, we study a specific model-based policy search framework that generates samples from a learned model and performs policy optimization with the samples. In particular, we analyze natural policy gradient with value equivalent models and study the property in finite-horizon MDPs with function approximations.

Denote the state value and state-action value by unrolling model $\hat{f}$ with policy π as $\hat{V}^\pi$ and $\hat{Q}^\pi$, respectively. That is, $\hat{V}^\pi = \sum_{h=1}^{k} \hat{r}(\hat{s}_h, \hat{a}_h)$, where $\hat{s}_h \sim \hat{f}(\cdot|\hat{s}_{h-1}, \hat{a}_{h-1})$, $\hat{a}_h \sim \pi(\cdot|\hat{s}_h)$, and k is

the task horizon. Here, the objective $J(\pi) = \mathbb{E}_{s_0}\left[V^{\pi}(s_0)\right]$ and $\hat{J}(\pi_{\theta_t})$ is estimated with B samples of $\hat{V}^{\pi}$. Then the gradient estimation is

$$\nabla_{\theta}\hat{J}(\pi_{\theta}) = \frac{1}{B}\sum_{b=1}^{B}\hat{Q}^{\pi}(s_b, a_b)\nabla_{\theta}\log\pi_{\theta}(a_b|s_b).$$

For natural gradient $\tilde{\nabla}_{\theta}J(\theta_t)$, a Fisher information matrix is adopted for update $\theta_{t+1} - \theta_t = \eta\tilde{\nabla}_{\theta}J(\theta_t)$, where

$$\tilde{\nabla}_{\theta}J(\theta) = F(\theta)^{-1}\nabla_{\theta}J(\theta) = F(\theta)^{-1}\mathop{\mathbb{E}}_{s\sim d_{\mu_1}^{\pi_\theta}}\mathop{\mathbb{E}}_{a\sim\pi_\theta}\left[A^{\pi_\theta}(s,a)\nabla_{\theta}\log\pi_{\theta}(a|s)\right].$$

Assumption 1. *V_{π_θ} is L-smooth in θ.*

This assumption holds when the reward r and transition function f are both Lipschitz continuous and smooth. Such assumptions are also consider by many previous works [2, 16, 11], and we give examples in Appendix 4.2 for Assumption 1 to hold.

Theorem 1. *Assume for all $s \in \mathcal{S}$ and $a \in \mathcal{A}$ that $\log\pi_{\theta}(a|s)$ is a ι-smooth function of θ. For $\eta \le \frac{1}{L}$,*

$$\min_{t\in[T]} J(\pi^*) - J(\pi_{\theta_t}) \le \frac{k\log|\mathcal{A}|}{T\eta} + \frac{k\iota}{T(1-L\eta)}\left(\mathbb{E}[J(\pi_{\theta_T}) - J(\pi_{\theta_1})] + \eta\sum_{t=1}^{T}(b_t + \frac{v_t}{2})\right),$$

where the gradient bias b_t and the upper bound v_t of gradient variance are defined by $b_t = \left\|\nabla_{\theta}J(\pi_{\theta_t}) - \mathbb{E}[\nabla_{\theta}\hat{J}(\pi_{\theta_t})]\right\|_2$ and $v_t = \mathbb{E}\left[\left\|\nabla_{\theta}\hat{J}(\pi_{\theta_t}) - \mathbb{E}[\nabla_{\theta}\hat{J}(\pi_{\theta_t})]\right\|_2^2\right]$.

Proof. See Appendix 4.1. □

What remains is to bound b_t and v_t in the convergence rate.

2.2 Online VE Model Learning

For convenience, we assume a bounded gradient variance. A wide range of RL problems can be contained with this assumption. See examples in Appendix. 4.2.

Assumption 2. *There exists an absolute constant $c \ge 0$ such that $v_t \le c^2/B$ for all $t \in [T]$.*

We first focus on the likelihood ratio gradient estimator that leverages the score function to calculate gradient. Let $\mathcal{T}_{\pi}^{k}$ be k applications of the policy's Bellman operator $\mathcal{T}_{\pi}$, defined as $\mathcal{T}_{\pi}[v](s) = \mathbb{E}[r(s,a) + v(s')|f^*, \pi]$. Similarly, $\hat{\mathcal{T}}_{\pi}$ is the Bellman operator induced by model $\hat{f}$ and policy π.

The gradient bias can be represented as $b_t = \|V^{\pi_t} - \hat{\mathcal{T}}_{\pi_t}^{k}V^{\pi_t}\|$, which corresponds to an order-k value equivalence loss [7] w.r.t. a single policy, *i.e.*, $\Pi_t = \pi_t$. Different from pure model-free algorithms, in MBPG (Algorithm 1) multiple policy gradient updates are performed with data generated from a learned model so that the sample complexity is reduced. In other words, the learned model $\hat{f}_t$ continues to have an impact on the following n policy updates. Thus, simply setting $\hat{f}_t = \operatorname{argmin}_{\hat{f}} b_{t-1}$ only guarantees a low-bias gradient update from π_{t-1} to π_t, leading to potentially large bias for the following $n-1$ policy gradients.

Formally, in the convergence rate Theorem 1, we need the accumulate $\sum_{t=1}^{T} b_t$ to be small. That is, the true objective for the model learner $\mathcal{R}$ is to minimize the following regret REG_T

$$\min_{\hat{f}_1,\ldots,\hat{f}_{\lfloor\frac{T}{k}\rfloor k}}\sum_{t=1}^{T}\mathcal{L}_t\left(\left((s,a),\hat{V}^{\pi}\right);\hat{f}_{\lfloor\frac{t}{k}\rfloor k}\right) \triangleq \min_{\hat{f}_1,\ldots,\hat{f}_{\lfloor\frac{T}{k}\rfloor k}}\sum_{t=1}^{T}\left\|V^{\pi_t} - \hat{\mathcal{T}}_{\pi_t}^{k}V^{\pi_t}\right\|,$$

where $\hat{\mathcal{T}}_{\pi_t}$ is the Bellman operator induced by model $\hat{f}_{\lfloor\frac{t}{k}\rfloor k}$.

We adopt the notation from [12] to define the sequential Rademacher complexity $\mathfrak{R}_T(\mathbb{L}; \boldsymbol{x}, \boldsymbol{y})$ of the loss function class $\mathbb{L} = \{(x, y) \mapsto \mathcal{L}((x, y); \hat{f})\}$. Here, x and y are training input (s, a) and predictions $\hat{V}^\pi$, respectively. Denote a sequence of Rademacher random variables $\epsilon = (\epsilon_1, \cdots, \epsilon_T)$. For any $\overline{\mathcal{X}}$-valued tree $\boldsymbol{x}$ and any $\mathcal{Y}$-valued tree $\boldsymbol{y}$, $\mathfrak{R}_T(\mathbb{L}; \boldsymbol{x}, \boldsymbol{y})$ is defined as

$$\mathfrak{R}_T(\mathbb{L}; \boldsymbol{x}, \boldsymbol{y}) \triangleq \mathop{\mathbb{E}}_{\epsilon, \xi} \left[\sup_{\mathcal{L} \in \mathbb{L}} \sum_{t=1}^{T} \epsilon_t \mathcal{L}((\boldsymbol{x}(\epsilon), \xi_t), \boldsymbol{y}(\epsilon)) \right].$$

Further define $\mathfrak{R}_T(\mathbb{L}) = \sup_{\boldsymbol{x}, \boldsymbol{y}} \mathfrak{R}_T(\mathbb{L}; \boldsymbol{x}, \boldsymbol{y})$. The model learner gives predictions based on $\mathcal{R}(\{x_i, y_i\}_{i=1}^{t-1})$ over the model space.

Leveraging standard results from [12], we can show the existence of an online model learner $\mathcal{R}$ that gives $\sum_{t=1}^{T} b_t \leq 2n\mathfrak{R}_T(\mathbb{L})$. Typically, we have the Rademacher complexity $\mathfrak{R}_T(\mathbb{L})$ bounded by $\tilde{O}(\sqrt{T})$, which is desired in the convergence rate theorems. Next, we discuss several such designs.

General model learners. We first provide a learner in general settings which maintains the policy cover $\Pi_t = \{\pi_i\}_{i=1}^{t-1}$. Define the model learning error at iteration t as $\delta_t = \sum_{i=1}^{t-1} \mathbb{E}_{s \sim d^{\pi_i}} \left\| (\mathcal{T}_{\pi_t}^k - \hat{\mathcal{T}}_{\pi_t}^k) V^{\pi_i} \right\|$, where $\hat{\mathcal{T}}_{\pi_t}$ is induced by model $\hat{f}_{\lfloor \frac{t}{n} \rfloor n}$.

Proposition 1. *With model learning error $\delta = \max_{t \in [1,T]} \delta_t$,*

$$\textit{REG}_T \leq (\delta + k)\sqrt{T(1 + k^2)\log(1 + k^2)}. \tag{1}$$

Suppose we have access to an offline optimization oracle that gives a finite constant σ, then regret $\textit{REG}_T \leq O(\sqrt{T \log T})$.

Proof. See Appendix 4.3. □

Selecting past policies on the policy improvement path as Π_t has shown its effectiveness in recent implementation [7], which further improves the state-of-the-art MuZero performance on Atari tasks.

Online convex learning. Despite the bound developed in general settings, for linear function approximations, the model learning process becomes an online *convex* optimization problem, where online gradient descent suffices to obtain $O(\sqrt{T})$ regret. In this case, we can simply leverage *iterative* model learning algorithms that set $\Pi_t = \pi_{t-1}$ and minimize $\delta_t = \|V^{\pi_{t-1}} - \hat{\mathcal{T}}_{\pi_{t-1}}^k V^{\pi_{t-1}}\|$. When the underlying transition of a high-dimensional MDP can be linearly modeled, we may also first convert the states to an abstract space to enjoy the convex property.

Extensions in nonconvex settings. Several extensions of Proposition 4.3 can be obtained in nonconvex settings, *e.g.*, when nonlinear function approximation is used in MDPs with complex transitions or large state space. In non-convex online learning, a heuristic yet provable algorithm is *Follow the Perturbed Leader* (FTPL). Similar with the FTL-style learner, FTPL adds a perturbation noise to the loss: $\delta_t = \sum_{i=1}^{t-1} \mathbb{E}_{s \sim d^{\pi_i}} \left\| (\mathcal{T}_{\pi_t}^k - \hat{\mathcal{T}}_{\pi_t}^k) V^{\pi_i} \right\| + \sigma_t$, where each coordinate of σ_t is sampled from $\text{Exp}(\lambda_t)$, the exponential distribution with parameter λ_t.

For non-oblivious loss which is not adversarially chosen, *e.g.*, the model learning loss, we may expect it predictable from the past. Thus, minimizing $\delta_t = \sum_{i=1}^{t-1} \mathbb{E}_{s \sim d^{\pi_i}} \left\| (\mathcal{T}_{\pi_t}^k - \hat{\mathcal{T}}_{\pi_t}^k) V^{\pi_i} \right\| + M_t + \sigma_t$ with a predictor M_t might provide useful information and the resulting optimistic algorithms can lead to tighter regret bound. For example, we may simply set $M_t = \mathcal{L}_{t-1}$ to emphasise the policy similarities between consecutive iterations, or importance weighted past losses $M_t = \frac{1}{t-1} \sum_{i=1}^{t-1} \alpha_i \mathcal{L}_i$ for fading memory statistics, or combined with additional losses given by random policies.

We show that FTPL and OFTPL achieve $O(\sqrt{T})$ regret with access to an offline optimization oracle.

Proposition 2. *Denote the (α, β)-approximate optimization oracle of function g as $\mathcal{O}_{\alpha,\beta}(g)$ in the sense that if $x^* = \mathcal{O}_{\alpha,\beta}(g)$, then $g(x^*) - \langle \sigma, x^* \rangle \leq \inf_x f(x) - \langle \sigma, x \rangle + \alpha + \beta \|\sigma\|_1$. Suppose the losses encountered by the model learner are l-Lipschitz w.r.t l_1 norm. For $\lambda = O(T^{-\frac{1}{2}})$, $\alpha = O(T^{-\frac{1}{2}})$, and $\beta = O(T^{-1})$, we have for FTPL and OFTPL that $\mathbb{E}[\textit{REG}_T] \leq O(\sqrt{T})$.*

All the above propositions guarantee an $O(T^{-\frac{1}{2}})$ rate of convergence in Theorem 1.

Discussions. For more general settings including pathwise gradient estimator and function approximations without natural gradients, we might use the analysis above to policy-aware or gradient-aware model learning [1, 4] that directly learns a model for accurate gradient estimation by imposing the score magnitude as an additional weight.

When the task horizon k is large, model rollout can lead to high variance and bias due to compounding error. In this case, we can learn an additional (non)parametric value function v and replace $\hat{V}^\pi$ with the k'-step bootstrapping $\hat{V}^\pi_{\text{bs}} = \sum_{h=1}^{k'} \hat{r}(\hat{s}_h, \hat{a}_h) + v(\hat{s}_{k'})$.

3 Model-Based Planning

Another important usage of models is to perform model-based planning. Much attention has been paid to the planning with value equivalent models, such as Predictron, MuZero, VPN, and PVE.

3.1 Case Study of MuZero

MuZero [14] is a modern architecture that shows the power of value equivalence principle in model-based planning. The agent maintains a predictive model that outputs abstract state transitions, values, and rewards, then the action returned by MCTS (with upper confidence bound applied to trees) is executed in the real MDPs. The two stages are carried out iteratively.

Grimm et al. [7] demonstrates the close connection between the VE loss and the MuZero objective. Specifically, MuZero's per-state model loss can be expressed as:

$$l(s_h) = \sum_{i=0}^{k} (V_{h+i} - v(z_h^i))^2 + (r_{h+i} - \hat{r}(z_h^i))^2,$$

where $V_{h+i} = r_{h+i} + \cdots + r_{h+i+k'-1} + v(z^0_{h+i+k'})$ is the k'-step bootstrapping. z_h^i is the abstract state after i step model rollout from the real state s_h. $s_{h:H}$, $a_{h:H}$, and $r_{h:H}$ are the real trajectory.

It is shown that the MuZero model loss upper bounds the order-k VE loss, *i.e.*, $\|V^{\pi_t} - \hat{\mathcal{T}}^k_{\pi_t} V^{\pi_t}\|^2_{d_\pi} \leq c\,\mathbb{E}_{d_\pi}[l(s)]$. Optimizing the MuZero model is equivalent to finding a VE model w.r.t. $\Pi_t = \pi_{t-1}$.

Since experiments in [7] show that even better results can be obtained by adding VE loss induced by past policies to MuZero, we investigate why $\Pi_t = \{\pi_i\}_{i=1}^{t-1}$ is preferred over $\Pi_t = \{\pi_{t-1}\}$.

UCB1 is an algorithm for multi-armed bandit that achieves logarithm regret with the number of actions taken. When applied to MCTS (UCT [10] or PUCT [13]), convergence to the optimal policy can be shown, if given access to the accurate payoffs. However, learning a unified model for value prediction is exactly the objective of MuZero, which results in iterative executions of tree search and model update. Thus, to derive the convergence of MuZero, we also need the predicted model value $\hat{\mathcal{T}}^k_{\pi_t} V^{\pi_t}$ to converge to the true MDP as iteration t increases, otherwise it can get stuck at local optima. This can be handled by online model learning.

Formally, for an optimal arm (action) with reward μ^* at a given state, define the regret to be the loss caused by the policy not always playing the best arm at the end of each iteration (T in total), *i.e.*, $\text{REG}_T = T\mu^* - \mathbb{E}[\sum_{t=1}^{T} \mu_{it}]$.

Proposition 3. *Denote the number of MCTS runs with model $\hat{f}_t$ in a single iteration t as n. Suppose $n = O(T^2)$. For MuZero with model learner $\mathcal{R}$, denote the Rademacher complexity $\mathfrak{R}_T(L)$ for the class L where $l \in L$. Then the regret REG_T is bounded by*

$$REG_T \leq O(\mathfrak{R}_T) + O(\sqrt{T}).$$

Proof. See Appendix 4.5. □

It is typical for online model learners that $\mathfrak{R}_T$ is bounded by $\tilde{O}(\sqrt{T})$. For example, define the model loss as $\sum_{\pi \in \Pi_t} \mathbb{E}_{d_\pi}[l(s)]$ and $\Pi_t = \{\pi_i\}_{i=1}^{t-1}$.

Algorithm 1 Model-Based Policy Gradient

Initialize policy π_0, model $\hat{f}$, $t = 1$
for $\lfloor \frac{T}{n} \rfloor n$ iterations **do**
  Execute policy π_{t-1} in the real MDP
  Update VE model $\hat{f}_t$ (*e.g.*, by minimizing δ_t)
  for n gradient updates **do**
    Calculate gradient $\nabla_\theta \hat{J}(\pi_{t-1})$
    Perform PG and obtain updated π_t
    $t \leftarrow t + 1$
  end for
end for

Algorithm 2 Model-Based Planning

Initialize policy π_0, model $\hat{f}$, $t = 1$
for $\lfloor \frac{T}{n} \rfloor n$ iterations **do**
  Execute policy π_{t-1} in the real MDP
  Update VE model $\hat{f}_t$
  for n planning iterations **do**
    Perform model-based planning (*e.g.*, MCTS)
    $t \leftarrow t + 1$
  end for
  Return π_t
end for

3.2 Planning in General Settings

Note that the UCB algorithms used in MCTS and MuZero can only be applied to independent arms. For more general settings such as nonlinear models, recent works [3] have shown pessimistic results for *global* convergence. Specifically, Dong et al. shows that the model complexity measure induced for nonlinear models (*e.g.*, Eluder dimension) cannot be polynomially bounded even for one-layer neural networks. It is thus suggested to find an (ϵ_g, ϵ_h)-approximate local optimal policy π^* and use local regret instead as the evaluation metric, defined as

$$\mathrm{REG}^{\mathrm{LC}}_{\epsilon_g,\epsilon_h}(T) = \sum_{t=1}^{T} \left(\sup_{\pi^* \in \mathfrak{C}_{\epsilon_g,\epsilon_h}} V^{\pi^*} - V^{\pi_t} \right)$$

Proposition 4. *Define the model loss* $l_t = \|V^{\pi_{t-1}} - \hat{V}^{\pi_{t-1}}\| + \|V^{\pi_t} - \hat{V}^{\pi_t}\|$ *and* $\mathfrak{R}_T(L)$ *the Rademacher complexity for class* L*, where* $l \in L$*. Under Lipschitz assumptions, we have for the* $(\epsilon, 6\sqrt{\zeta\epsilon})$*-regret that*

$$\mathit{REG}^{\mathit{LC}}_{\epsilon,6\sqrt{\zeta\epsilon}}(T) \leq O(\sqrt{T\mathfrak{R}_T}).$$

Notably, the loss l_t contains the prediction error w.r.t. both π_{t-1} at the previous iteration and π_t, the *upcoming* policy. The intuition is to optimize the model so that monotonic policy improvement can be obtained to achieve the local optima within the convex hull. The simplest way to meet such condition is to give accurate predictions for successive policy updates. This motivates the usage of VE objectives with online model learners, such as FTL-style algorithms.

4 Proofs

4.1 Proof of Theorem 1

Proof. Denote $\beta_t = \frac{\theta_{t+1}-\theta_t}{\eta} = \tilde{\nabla}_\theta \hat{J}(\theta_t)$. By Assumption 1, we have

$$\begin{aligned} J(\pi_{\theta_{t+1}}) - J(\pi_{\theta_t}) &\geq \nabla_\theta J(\pi_{\theta_t})^\top (\theta_{t+1} - \theta_t) - \frac{L}{2}\|\theta_{t+1} - \theta_t\|_2^2 \\ &= \eta \nabla_\theta J(\pi_{\theta_t})^\top \beta_t - \frac{L\eta^2}{2}\|\beta_t\|_2^2. \end{aligned} \tag{2}$$

Rewrite the exact gradient $\nabla_\theta J(\pi_{\theta_t})$ as

$$\nabla_\theta J(\pi_{\theta_t}) = \Big(\nabla_\theta J(\pi_{\theta_t}) - \mathbb{E}[\nabla_\theta \hat{J}(\pi_{\theta_t})]\Big) - \Big(\nabla_\theta \hat{J}(\pi_{\theta_t}) - \mathbb{E}[\nabla_\theta \hat{J}(\pi_{\theta_t})]\Big) + \nabla_\theta \hat{J}(\pi_{\theta_t}).$$

Then we bound $\nabla_\theta J(\pi_{\theta_t})^\top \beta_t$ in Eq. (2) by bounding the resulting three terms.

$$\left|\Big(\nabla_\theta J(\pi_{\theta_t}) - \mathbb{E}[\nabla_\theta \hat{J}(\pi_{\theta_t})]\Big)^\top \beta_t\right| \leq \|\beta_t\|_2 \|\nabla_\theta J(\pi_{\theta_t}) - \mathbb{E}[\nabla_\theta \hat{J}(\pi_{\theta_t})]\|_2 = \|\beta_t\|_2 b_t \tag{3}$$

$$\Big(\nabla_\theta \hat{J}(\pi_{\theta_t}) - \mathbb{E}[\nabla_\theta \hat{J}(\pi_{\theta_t})]\Big)^\top \beta_t \leq \frac{\|\nabla_\theta \hat{J}(\pi_{\theta_t}) - \mathbb{E}[\nabla_\theta \hat{J}(\pi_{\theta_t})]\|_2^2}{2} + \frac{\|\beta_t\|_2^2}{2} \tag{4}$$

$$\nabla_\theta \hat{J}(\pi_{\theta_t})^\top \beta_t \geq \|\beta_t\|_2^2, \tag{5}$$

where Eq. (5) holds due to the fact that $\Big(\theta_{t+1} - (\theta_t + \eta \nabla_\theta \hat{J}(\theta_t))\Big)^\top (\theta_{t+1} - \theta_t) \leq 0$.

Thus, we can bound Eq. (2) by

$$J(\pi_{\theta_{t+1}}) - J(\pi_{\theta_t}) \geq \eta \left(-\|\beta_t\|_2 b_t - \frac{\|\nabla_\theta \hat{J}(\pi_{\theta_t}) - \mathbb{E}[\nabla_\theta \hat{J}(\pi_{\theta_t})]\|_2^2}{2} + \frac{\|\beta_t\|_2^2}{2}\right) - \frac{L\eta^2}{2}\|\beta_t\|_2^2. \tag{6}$$

By taking expectation in Eq. (6), we have

$$\begin{aligned} \left(\frac{\eta}{2} - \frac{L\eta^2}{2}\right)\mathbb{E}[\|\beta_t\|_2^2] &\leq \mathbb{E}[J(\pi_{\theta_{t+1}}) - J(\pi_{\theta_t})] + \eta\, \mathbb{E}[\|\beta_t\|_2] b_t + \frac{\eta}{2} v_t \\ &\leq \mathbb{E}[J(\pi_{\theta_{t+1}}) - J(\pi_{\theta_t})] + b_t + \frac{\eta}{2} v_t \end{aligned} \tag{7}$$

We have for $\eta \leq \frac{1}{L}$ that

$$\mathbb{E}[\|\beta_t\|_2^2] \leq 2(\eta - L\eta^2)^{-1} \left(\mathbb{E}[J(\pi_{\theta_{t+1}}) - J(\pi_{\theta_t})] + \eta b_t + \frac{\eta}{2} v_t\right). \tag{8}$$

By smoothness, we have

$$\log \frac{\pi_{t+1}(a|s)}{\pi_t(a|s)} - \nabla_\theta \log \pi_t(a|s)(\theta_{t+1} - \theta_t) \geq -\frac{\iota}{2}\|\theta_{t+1} - \theta_t\|_2^2. \tag{9}$$

Then we obtain by the definition of KL divergence that

$$\begin{aligned} \mathop{\mathbb{E}}_{s\sim d^{\pi^*}}[D_{\mathrm{KL}}(\pi^*||\pi_t) - D_{\mathrm{KL}}(\pi^*||\pi_{t+1})] &= \mathop{\mathbb{E}}_{s\sim d^{\pi^*}} \mathop{\mathbb{E}}_{a\sim\pi^*}\left[\log \frac{\pi_{t+1}(a|s)}{\pi_t(a|s)}\right] \\ &\geq \mathop{\mathbb{E}}_{s\sim d^{\pi^*}} \mathop{\mathbb{E}}_{a\sim\pi^*}\left[\nabla_\theta \log \pi_t(a|s)(\theta_{t+1} - \theta_t) - \frac{\iota}{2}\|\theta_{t+1} - \theta_t\|_2^2\right] \\ &= \mathop{\mathbb{E}}_{s\sim d^{\pi^*}} \mathop{\mathbb{E}}_{a\sim\pi^*}\left[\eta A^{\pi_t}(s,a) - \frac{\iota\eta^2}{2}\|\beta_t\|_2^2\right] \\ &= \frac{1}{h}\eta\left(J(\pi^*) - J(\pi_t)\right) - \frac{\iota\eta^2}{2}\mathbb{E}[\|\beta_t\|_2^2]. \end{aligned} \tag{10}$$

where the last equality follows the performance difference lemma [9].

Thus,

$$
\begin{aligned}
&\min_{t\in[T]} J(\pi^*) - J(\pi_{\theta_t}) \\
&\leq \frac{1}{T}\sum_{t=1}^{T}\left(J(\pi^*) - J(\pi_{\theta_t})\right) \\
&\leq \frac{k}{T\eta}\sum_{t=1}^{T}\mathop{\mathbb{E}}_{s\sim d^{\pi^*}}[D_{\mathrm{KL}}(\pi^*||\pi_t) - D_{\mathrm{KL}}(\pi^*||\pi_{t+1})] + \frac{\iota\eta^2}{2}\,\mathbb{E}[\|\beta_t\|_2^2] \\
&\leq \frac{k\,\mathbb{E}_{s\sim d^{\pi^*}}[D_{\mathrm{KL}}(\pi^*||\pi_1)]}{T\eta} + \frac{\iota k}{T(1-L\eta)}\left(\mathbb{E}[J(\pi_{\theta_T}) - J(\pi_{\theta_1})] + \eta\sum_{t=1}^{T}(b_t + \frac{v_t}{2})\right) \\
&\leq \frac{k\log|\mathcal{A}|}{T\eta} + \frac{k\iota}{T(1-L\eta)}\left(\mathbb{E}[J(\pi_{\theta_T}) - J(\pi_{\theta_1})] + \eta\sum_{t=1}^{T}(b_t + \frac{v_t}{2})\right).
\end{aligned}
$$

□

4.2 Examples

We first provide the a possible condition for Assumption 1 to hold.

Example 1. *(Bastani, 2021, Lemma D.2). Denote L_h as the Lipschitz constant for function h and $\bar{L}_h = \max\{L_h, L_{\nabla h,1}\}$. Then ∇V_{π_θ} is L-Lipschitz, where $L = 44H^5\bar{L}_r\bar{L}_f^{4H}$. Particularly, $\nabla_\theta V_{\pi_\theta}$ is Lipschitz continuous in θ with Lipschitz constant $24H^5\bar{L}_r\bar{L}_f^{4H}$.*

We also give an example setting that satisfies Assumption 2.

Example 2. *With function approximations, for a stochastic model with form $\hat{s}_{h+1} = \hat{f}(\hat{s}_h, \hat{a}_h) + \xi_h$, where $\xi_h \sim p(\xi)$ and $p(\xi)$ is σ_ξ-subgaussian. Assume a Lipschitz continuous $\hat{f}$ and $\nabla\hat{r}$. For initial state distribution $\|s_0 - \mathbb{E}[s_0]\|_2 \leq \|\mathcal{S}\|_2$ and $\|a\|_2 \leq 1$, we have that $v_r \leq c_t^2/B$ where*

$$
c_t = \left(\frac{L_{\nabla\hat{r}}(L_{\hat{f}} + \sigma_\xi)}{L_{\hat{f}} - 1} + L_{\nabla\hat{r}}\|\mathcal{S}\|_2\right)\frac{2L_{\hat{f}}(L_{\hat{f}}^k - 1)}{L_{\hat{f}} - 1} + kL_{\nabla\hat{r}}. \tag{11}
$$

Setting c as the largest c_t satisfies Assumption 2.

Proof. First, we have

$$
\begin{aligned}
\|\hat{s}_{h+1} - \mathbb{E}[\hat{s}_{h+1}]\|_2 &= \left\|\hat{f}(\hat{s}_h, \hat{a}_h) + \xi_h - \mathbb{E}[\hat{f}(\hat{s}_h, \hat{a}_h)]\right\|_2 \\
&\leq L_{\hat{f}}\sqrt{\|\hat{s}_h - \mathbb{E}[\hat{s}_h]\|_2^2 + 1} + \sigma_\xi \\
&\leq L_{\hat{f}}\,\|\hat{s}_h - \mathbb{E}[\hat{s}_h]\|_2 + L_{\hat{f}} + \sigma_\xi.
\end{aligned} \tag{12}
$$

Since $\|s_0 - \mathbb{E}[s_0]\|_2 \leq \|\mathcal{S}\|_2$, we have for $h \geq 2$ that

$$
\|\hat{s}_h - \mathbb{E}[\hat{s}_h]\|_2 \leq \left(L_{\hat{f}} + \sigma_\xi\right)\frac{L_{\hat{f}}^{h-2} - 1}{L_{\hat{f}} - 1} + L_{\hat{f}}^{h-1}\|\mathcal{S}\|_2
$$

.

Besides,

$$\begin{aligned}\left\|\nabla_\theta \hat{J}(\pi_{\theta_t}) - \mathbb{E}[\nabla_\theta \hat{J}(\pi_{\theta_t})]\right\|_2 &= \left\|\sum_{h=0}^{k} \nabla \hat{r}(\hat{s}_h, \hat{a}_h) - \mathbb{E}[\nabla \hat{r}(\hat{s}_h, \hat{a}_h)]\right\|_2 \\ &\leq \sum_{h=0}^{k} L_{\nabla \hat{r}} \left(\|\hat{s}_h - \mathbb{E}[\hat{s}_h]\|_2 + 1\right) \\ &\leq \left(\frac{L_{\nabla \hat{r}}(L_{\hat{f}} + \sigma_\xi)}{L_{\hat{f}} - 1} + L_{\nabla \hat{r}} \|\mathcal{S}\|_2\right) \frac{2L_{\hat{f}}(L_{\hat{f}}^k - 1)}{L_{\hat{f}} - 1} + k L_{\nabla \hat{r}}.\end{aligned} \tag{13}$$

□

4.3 Proof of Proposition 1

Proof.

$$\begin{aligned}\left\|V^{\pi_t} - \hat{\mathcal{T}}_{\pi_t}^k V^{\pi_t}\right\|^2 &= \left\|(\mathcal{T}_{\pi_t}^k - \hat{\mathcal{T}}_{\pi_t}^k) V^{\pi_t}\right\|^2 \\ &\leq \|V^{\pi_t}\|_{\mathcal{V}_{t-1}^{-1}}^2 \left(\sum_{i=1}^{t-1} \mathop{\mathbb{E}}_{s \sim d^{\pi_i}} \left\|(\mathcal{T}_{\pi_t}^k - \hat{\mathcal{T}}_{\pi_t}^k) V^{\pi_i}\right\|^2\right) \\ &= (\delta_t^2 + k^2) \|V^{\pi_t}\|_{\mathcal{V}_{t-1}^{-1}}^2\end{aligned} \tag{14}$$

where $\mathcal{V}_t = 1 + \sum_{i=1}^{t} \mathbb{E}_{s \sim d^{\pi_i}}[V^{\pi_i}(s)^2]$. By writing it in a recursive form,

$$\begin{aligned}\mathcal{V}_t &= \mathcal{V}_{t-1} + \mathop{\mathbb{E}}_{s \sim d^{\pi_t}}[V^{\pi_i}(s)^2] \\ &= \mathcal{V}_{t-1} \left(1 + \mathcal{V}_{t-1}^{-1} \mathop{\mathbb{E}}_{s \sim d^{\pi_t}}[V^{\pi_i}(s)^2]\right)\end{aligned} \tag{15}$$

Then we have

$$\begin{aligned}\log \mathcal{V}_t &= \log \mathcal{V}_{t-1} + \log \left(1 + \mathcal{V}_{t-1}^{-1} \mathop{\mathbb{E}}_{s \sim d^{\pi_t}}[V^{\pi_i}(s)^2]\right) \\ &\geq \log \mathcal{V}_{t-1} + \frac{\mathcal{V}_{t-1}^{-1} \mathbb{E}_{s \sim d^{\pi_t}}[V^{\pi_i}(s)^2]}{1 + \mathcal{V}_{t-1}^{-1} \mathbb{E}_{s \sim d^{\pi_t}}[V^{\pi_i}(s)^2]} \\ &\geq \log \mathcal{V}_{t-1} + \frac{\mathcal{V}_{t-1}^{-1} \mathbb{E}_{s \sim d^{\pi_t}}[V^{\pi_i}(s)^2]}{1 + k^2}\end{aligned} \tag{16}$$

Thus,

$$\begin{aligned}\sum_{t=1}^{T} \|V^{\pi_t}\|_{\mathcal{V}_{t-1}^{-1}} &\leq \sqrt{T} \sqrt{\sum_{t=1}^{T} \|V^{\pi_t}\|_{\mathcal{V}_{t-1}^{-1}}^2} \\ &\leq \sqrt{T} \sqrt{\sum_{t=1}^{T} (1 + k^2) (\log \mathcal{V}_t - \log \mathcal{V}_{t-1})} \\ &= \sqrt{T} \sqrt{(1 + k^2) \log \mathcal{V}_T} \\ &\leq \sqrt{T (1 + k^2) \log(1 + k^2)}\end{aligned} \tag{17}$$

Plugging Eq. (17) into Eq, (14) gives the result.

□

4.4 Proof of Proposition 2

Proof. For regret REG_T, we have

$$\begin{aligned}
\text{REG}_T &= \sum_{t=1}^{T} \mathcal{L}_t(\hat{f}_{\lfloor \frac{T}{n} \rfloor n}) \\
&= \sum_{t=1}^{T} \left[\mathcal{L}_t(\hat{f}_{\lfloor \frac{T}{n} \rfloor n}) - \mathcal{L}_t(\hat{f}_{t+1}) \right] + \sum_{t=1}^{T} \mathcal{L}_t(\hat{f}_{t+1}) \\
&\leq l \sum_{t=1}^{T} \|\hat{f}_{\lfloor \frac{T}{n} \rfloor n} - \hat{f}_{t+1}\|_1 + \sum_{t=1}^{T} \mathcal{L}_t(\hat{f}_{t+1}) \\
&= l \sum_{j=1}^{\lfloor \frac{T}{n} \rfloor n} \sum_{i=1}^{n} \|\hat{f}_{jn} - \hat{f}_{jn+i}\|_1 + \sum_{t=1}^{T} \mathcal{L}_t(\hat{f}_{t+1}) \\
&\leq l \sum_{j=1}^{\lfloor \frac{T}{n} \rfloor n} \sum_{i=1}^{n} \|\hat{f}_{jn} - \hat{f}_{jn+1}\|_1 + \ldots + \|\hat{f}_{jn+i-1} - \hat{f}_{jn+i}\|_1 + \sum_{t=1}^{T} \mathcal{L}_t(\hat{f}_{t+1}) \\
&= l \sum_{j=1}^{\lfloor \frac{T}{n} \rfloor n} \sum_{i=1}^{n} (n+1-i) \|\hat{f}_{jn+i-1} - \hat{f}_{jn+i}\|_1 + \sum_{t=1}^{T} \mathcal{L}_t(\hat{f}_{t+1}).
\end{aligned}$$

Note that we only optimize the model at iteration $\lfloor \frac{T}{n} \rfloor n$, and the virtual $\hat{f}_t$ at other iteration t is not what we have during training. It is introduced for the proof only.

By applying induction, we have for the expected regret

$$\begin{aligned}
\mathbb{E}\left[\sum_{t=1}^{T} \mathcal{L}_t(\hat{f}_{\lfloor \frac{T}{n} \rfloor n}) \right] &\leq l \sum_{t=1}^{T} \frac{n(n+1)}{2} \mathbb{E}[\|\hat{f}_t - \hat{f}_{t+1}\|_1] + \sum_{t=1}^{T} \mathbb{E}[\mathcal{L}_t(\hat{f}_{t+1})] \\
&\leq l \sum_{t=1}^{T} \frac{n(n+1)}{2} \mathbb{E}[\|\hat{f}_t - \hat{f}_{t+1}\|_1] + \frac{d(\beta T + D)}{\lambda} + \alpha T.
\end{aligned} \tag{18}$$

The stability term $\mathbb{E}[\|\hat{f}_t - \hat{f}_{t+1}\|_1]$ remains to be bounded. From Lemma 5 and Lemma 6 in [15], we have

$$\mathbb{E}[\|\hat{f}_t - \hat{f}_{t+1}\|_1] \leq 125\lambda D l d^2 + \frac{d\beta}{20\lambda l} + 2d\beta + \frac{\alpha}{20l}.$$

Plugging the above bound in Equation (18) gives the bound of expected regret.

$$\mathbb{E}\left[\sum_{t=1}^{T} \mathcal{L}_t(\hat{f}_{\lfloor \frac{T}{n} \rfloor n}) \right] \leq O\left(T\lambda D l^2 d^2 n^2 + dl\beta + \frac{d(\beta T + D)}{\lambda} + \alpha T \right).$$

For $\lambda = O(T^{-\frac{1}{2}})$, $\alpha = O(T^{-\frac{1}{2}})$, and $\beta = O(T^{-1})$, we have the $O(\sqrt{T})$ expect regret bound. $\square$

4.5 Proof of Proposition 3

Proof. Denote $a^*_{\hat{f}}$ as the optimal arm under $\hat{f}$ and $\mu^*_{\hat{f}}$ as the return by playing $a^*_{\hat{f}}$. Then we can decompose the regret by

$$\begin{aligned}
\text{REG}_T &= T\mu^* - \mathbb{E}[\sum_{t=1}^{T} \mu_{it}] \\
&= T\mu^* - \sum_{t=1}^{T} \mu^*_{\hat{f}_t} + \sum_{t=1}^{T} \mu^*_{\hat{f}_t} - \mathbb{E}[\sum_{t=1}^{T} \mu_{it}] \\
&\leq T\mu^* - \sum_{t=1}^{T} \mu^*_{\hat{f}_t} + O(T\sqrt{\frac{\log n}{n}}) \\
&\leq O(\mathfrak{R}_T) + O(\sqrt{T}),
\end{aligned} \tag{19}$$

where the first inequality follows the results in PUCT [13]. In the last inequality, the first term is due to the property of Rademacher complexity and the second term is because $n = O(T^2)$ and $O(T\sqrt{\frac{\log n}{n}}) \leq O(T\sqrt{\frac{\log(T^2)}{T^2}}) \leq O(\sqrt{T})$. □

References

[1] Romina Abachi, Mohammad Ghavamzadeh, and Amir-massoud Farahmand. Policy-aware model learning for policy gradient methods. *arXiv preprint arXiv:2003.00030*, 2020.

[2] Osbert Bastani. Sample complexity of estimating the policy gradient for nearly deterministic dynamical systems. In *International Conference on Artificial Intelligence and Statistics*, pages 3858–3869. PMLR, 2020.

[3] Kefan Dong, Jiaqi Yang, and Tengyu Ma. Provable model-based nonlinear bandit and reinforcement learning: Shelve optimism, embrace virtual curvature. *Advances in Neural Information Processing Systems*, 34, 2021.

[4] Pierluca D'Oro, Alberto Maria Metelli, Andrea Tirinzoni, Matteo Papini, and Marcello Restelli. Gradient-aware model-based policy search. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 3801–3808, 2020.

[5] Amir-massoud Farahmand. Iterative value-aware model learning. In *NeurIPS*, pages 9090–9101, 2018.

[6] Amir-massoud Farahmand, Andre Barreto, and Daniel Nikovski. Value-aware loss function for model-based reinforcement learning. In *Artificial Intelligence and Statistics*, pages 1486–1494. PMLR, 2017.

[7] Christopher Grimm, André Barreto, Gregory Farquhar, David Silver, and Satinder Singh. Proper value equivalence. *arXiv preprint arXiv:2106.10316*, 2021.

[8] Christopher Grimm, André Barreto, Satinder Singh, and David Silver. The value equivalence principle for model-based reinforcement learning. *arXiv preprint arXiv:2011.03506*, 2020.

[9] Sham Kakade and John Langford. Approximately optimal approximate reinforcement learning. In *In Proc. 19th International Conference on Machine Learning*. Citeseer, 2002.

[10] Levente Kocsis and Csaba Szepesvári. Bandit based monte-carlo planning. In *European conference on machine learning*, pages 282–293. Springer, 2006.

[11] Matteo Pirotta, Marcello Restelli, and Luca Bascetta. Policy gradient in lipschitz markov decision processes. *Machine Learning*, 100(2):255–283, 2015.

[12] Alexander Rakhlin, Karthik Sridharan, and Ambuj Tewari. Online learning: Stochastic, constrained, and smoothed adversaries. *Advances in neural information processing systems*, 24, 2011.

[13] Christopher D Rosin. Multi-armed bandits with episode context. *Annals of Mathematics and Artificial Intelligence*, 61(3):203–230, 2011.

[14] Julian Schrittwieser, Ioannis Antonoglou, Thomas Hubert, Karen Simonyan, Laurent Sifre, Simon Schmitt, Arthur Guez, Edward Lockhart, Demis Hassabis, Thore Graepel, et al. Mastering atari, go, chess and shogi by planning with a learned model. *Nature*, 588(7839):604–609, 2020.

[15] Arun Sai Suggala and Praneeth Netrapalli. Online non-convex learning: Following the perturbed leader is optimal. In *Algorithmic Learning Theory*, pages 845–861. PMLR, 2020.

[16] Lingxiao Wang, Qi Cai, Zhuoran Yang, and Zhaoran Wang. Neural policy gradient methods: Global optimality and rates of convergence. *arXiv preprint arXiv:1909.01150*, 2019.