Optimism is All You Need: Model-Based Imitation Learning From Observation Alone

Anonymous authors
Paper under double-blind review

Abstract

This paper studies Imitation Learning from Observations alone (ILFO) where the learner is presented with expert demonstrations that only consist of states encountered by an expert (without access to actions taken by the expert). This paper presents a provably efficient model-based framework `MobILE` to solve the ILFO problem. `MobILE` uses self-supervision towards (a) training a dynamics model and (b) designing an intrinsic reward signal for exploration. Using these ideas, `MobILE` carefully trades off exploration against imitation by integrating the idea of optimism in the face of uncertainty into the distribution matching imitation learning (IL) framework. We provide a unified analysis for `MobILE`, and demonstrate that `MobILE` enjoys strong performance guarantees for classes of MDP dynamics that satisfy certain well studied notions of complexity. We also show that the ILFO problem is *strictly harder* than the standard IL problem by reducing ILFO to a multi-armed bandit problem indicating that strategic exploration is *necessary* for solving ILFO efficiently. We complement these theoretical results with experimental simulations on benchmark OpenAI Gym tasks that indicate the efficacy of `MobILE`.

1 Introduction

Imitation Learning (IL) is a paradigm that allows for sample-efficient learning of sequential decision making policies, utilizing expert demonstrations consisting of states and actions. IL has been successfully applied to applications such as Natural Language Processing (Daumé et al., 2009), Compilers (Mendis et al., 2019), Robotics (Levine et al., 2015), navigation (Ziebart et al., 2008), *etc.*

This paper, instead, considers the *Imitation Learning from Observation Alone (ILFO)* setting. In ILFO, the learner is presented with sequences of states encountered by the expert, *without* access to the actions taken by the expert, meaning approaches based on a reduction to supervised learning (e.g., Behavior cloning (BC) (Ross & Bagnell, 2010), DAgger Ross et al. (2011b)) are not applicable. ILFO is more general and has potential for applications where learners and experts have different action spaces, applications like sim-to-real (Song et al., 2020; Desai et al., 2020).

Recently, Sun et al. (2019b) reduces the ILFO problem to a sequence of one-step distribution matching problems that results in obtaining a non-stationary policy. This approach is sample inefficient for longer horizon tasks since the algorithm does not reuse previously collected samples when solving the current sub-problem. Another line of work considers model-based methods to infer the expert's actions with either an inverse (Torabi et al., 2018) or a forward dynamics (Edwards et al., 2019) model; these recovered actions are then fed into an IL approach like BC to output the final policy. These works rely on strong assumptions that tend to be satisfied when the underlying Markov Decision Process (MDP) has deterministic transition dynamics. See related works for a detailed treatment.

We introduce `MobILE`, a model-based framework, to solve the ILFO problem. In contrast to existing model-based efforts, `MobILE` learns the forward transition dynamics model based on online interactions, and this is *a well-defined learnable object*. Importantly, `MobILE` *combines exploration with imitation* by utilizing ideas from self-supervised learning to interleave a model learning step with an intrinsic bonus-based, optimistic distribution matching step – a perspective, to the best of our knowledge, that has not been considered in the ILFO setting. At a high level, our theoretical results and experimental studies demonstrate that

Strategic exploration is necessary for solving ILFO reliably and efficiently.

Note that this paper's result extends the realm of partial information problems where optimism has been shown to be crucial in obtaining strong performance both in theory (e.g., E^3 (Kearns & Singh, 2002b), UCB (Auer et al., 2002)) and practice (e.g., RND (Burda et al., 2018)). This paper proves that incorporating optimism into the min-max IL framework (Ho & Ermon, 2016; Sun et al., 2019b) is *necessary* for both the theoretical foundations and empirical performance of ILFO.

Main contributions: `MobILE` (Algorithm 1) is a provably efficient, model-based framework for ILFO. `MobILE` can be instantiated with flexible implementation choices owing to its modular design; `MobILE` also presents strong empirical performance on benchmark OpenAI gym tasks.

1. `MobILE` uses self supervision by combining ideas from model-based learning and optimism for exploration with adversarial imitation learning. `MobILE` achieves global optimality with a regret bound growing sublinearly for classes of MDP dynamics that satisfy certain well studied notions of complexity. The key idea of `MobILE` is to use optimism to *trade-off imitation and exploration*.
2. We show that optimism-based exploration is *necessary* for solving ILFO by presenting a reduction from the multi-armed bandit problem to ILFO. This indicates that ILFO is *fundamentally harder* than IL where the learner has access to the expert's actions. Thus unlike classic methods like BC and DAgger, reductions to supervised learning are not sufficient to provably solve ILFO.
3. We instantiate `MobILE` with neural networks and present experimental results on benchmark OpenAI Gym tasks, indicating `MobILE` compares favorably to or outperforms existing approaches. Ablation studies indicate that optimism indeed boosts performance in practice.

1.1 Related Works

Imitation Learning (IL) has seen advances through two types of approaches: (a) behavior cloning (BC) (Pomerleau, 1989) which casts IL as supervised or full-information online learning (Ross & Bagnell, 2010; Ross et al., 2011b), or, (b) inverse RL (Ng & Russell, 2000; Abbeel & Ng, 2004; Finn et al., 2016; Ke et al., 2019; Ghasemipour et al., 2020), which involves minimizing various distribution divergences to solve the IL problem, either with the transition dynamics known (e.g., Ziebart et al. (2008)), or unknown (e.g., Ho & Ermon (2016)). `MobILE` does not assume knowledge of the transition dynamics, is model-based, and operates without access to the expert's actions.

Imitation Learning from Observation Alone (ILFO) Sun et al. (2019b) presents a model-free approach that outputs a non-stationary policy by reducing the ILFO problem into a sequence of min-max problems, one per time-step. This approach is sample inefficient, particularly with long horizons; in contrast, our paper learns a stationary policy using model-based approaches. Another line of work (Torabi et al., 2018; Edwards et al., 2019; Yang et al., 2019) relies on learning an estimate of expert action, often through the use of an inverse dynamics models, $P^e(a|s,s')$. Unfortunately, there are many settings where an inverse model is not always well defined. For instance, inverse dynamics can be well defined for deterministic MDP transition dynamics, whereas, this is not the case for stochastic transition dynamics. From Bayes rule, to define an inverse model, we need a state-wise prior distribution and the expert policy, i.e., $P^e(a|s,s') \propto P(s'|s,a)\rho(s)\pi^e(a|s)$. Thus to learn $P^e(a|s,s')$, we need training data with actions from π^e[1], which is missing in ILFO. Finally, Edwards et al. (2019)'s result applies only to MDPs with deterministic transitions and discrete actions. Refer to section 6 in Sun et al. (2019b) for a more detailed treatment. `MobILE` instead learns a forward dynamics model which is unique and well-defined for deterministic and stochastic transitions and works with discrete/continuous actions. Finally, Peng et al. (2018); Aytar et al. (2018); Schmeckpeper et al. (2020) solve ILFO using cost function engineering based on task-specific priors; `MobILE` doesn't require cost function engineering.

Model-Based RL (Sutton, 1990; Li & Todorov, 2004; Deisenroth & Rasmussen, 2011) has seen advances based on deep learning (Lampe & Riedmiller, 2014; Gu et al., 2016; Janner et al., 2019) which can be translated to ILFO owing to `MobILE`'s modularity. `MobILE` bears parallels to provably efficient model-based RL approaches including E^3 (Kearns & Singh, 2002a; Kakade et al., 2003), R-MAX (Brafman & Tennenholtz, 2001), UCBVI (Azar et al., 2017), and LC^3 (Kakade et al., 2020a) which utilize a bonus based approach to trade-off exploration against exploitation.

[1]off-policy learning is not possible either unless one has additional information about π^e, e.g., access to the likelihood $\pi^e(a|s)$.

2 SETTING

We consider episodic finite-horizon MDP $\mathcal{M} = \{\mathcal{S}, \mathcal{A}, P^\star, H, c, s_0\}$, where $\mathcal{S}, \mathcal{A}$ are the state, action space, $P^\star : \mathcal{S} \times \mathcal{A} \mapsto \mathcal{S}$ is the transition kernel, H the horizon, s_0 is a fixed initial state (note that we can handle a distribution over initial states), and c is the *state-dependent* cost function $c : \mathcal{S} \mapsto [0, 1]$.
Notations We denote $d_P^\pi \in \Delta(\mathcal{S} \times \mathcal{A})$ as the average state-action distribution of policy π under the transition kernel P, i.e., $d_P^\pi(s, a) := \frac{1}{H}\sum_{t=1}^H Pr(s_t = s, a_t = a|s_0, \pi, P)$, where $Pr(s_t = s, a_t = a|s_0, \pi, P)$ is the probability of reaching (s, a) at time step t starting from s_0 by following π under transition kernel P. We abuse notation and write $s \sim d_P^\pi$ to denote a state s is sampled from the state-wise distribution which marginalizes action over $d_P^\pi(s, a)$, i.e., $d_P^\pi(s) := \frac{1}{H}\sum_{t=1}^H Pr(s_t = s|s_0, \pi, P)$. For a given cost function $f : \mathcal{S} \mapsto [0, 1]$, $V_{P;f}^\pi$ denotes the expected total cost of π under transition P and cost function f. Similar to IL, in ILFO, the *ground truth cost c is unknown*. Instead, we can query the expert, denoted as $\pi^e : \mathcal{S} \mapsto \Delta(\mathcal{A})$, to provide state-only demonstrations $\tau = \{s_0, s_1 \dots s_H\}$, where $s_{t+1} \sim P^\star(\cdot|s_t, a_t)$ and $a_t \sim \pi^\star(\cdot|s_t)$.

Note that the expert demonstration always starts from s_0, which is the starting state distribution of the MDP $\mathcal{M}$. Unlike interactive IL methods (e.g., DAgger (Ross et al., 2011a) and AggreVaTe(D) (Ross & Bagnell, 2014; Sun et al., 2017)), this paper considers the non-interactive expert setting where the algorithm cannot query the expert to provide trajectories starting from any state that the learner visits.

The goal is to leverage expert's state-wise demonstrations to learn a policy π that performs as well as π^e in terms of optimizing the ground truth cost c, with polynomial sample complexity on problem parameters such as horizon, number of expert samples and online samples and underlying MDP's complexity measures (see section 4 for precise examples).

2.1 FUNCTION APPROXIMATION SETUP

Since the ground truth cost c is unknown, we utilize the notion of a function class $\mathcal{F} \subset \mathcal{S} \mapsto [0, 1]$ to define the costs. Furthermore, we use a model class $\mathcal{P} \subset \mathcal{S} \times \mathcal{A} \mapsto \Delta(\mathcal{S})$ to capture the ground truth transition $P^\star$. For the theoretical results in the paper, we assume realizability:

Assumption 1. *Assume that $\mathcal{F}$, $\mathcal{P}$ captures ground truth cost and transition, i.e., $c \in \mathcal{F}$, and $P^\star \in \mathcal{P}$.*

To permit generalization, we require $\mathcal{P}$ to have bounded complexity. For analysis simplicity, assume $\mathcal{F}$ is discrete (but exponentially large), and we require the sample complexity of any PAC algorithm to scale polynomially with respect to its complexity $\ln(|\mathcal{F}|)$. The $\ln|\mathcal{F}|$ complexity can be replaced to bounded conventional complexity measures such as Rademacher complexity and covering number.

For the true model $P^\star$ and model class $\mathcal{P}$ we make the following structural assumption that the transition $P^\star$ is determined by a nonlinear deterministic function with additive Gaussian noise, i.e.,

$$s' \sim P(\cdot|s, a), \text{ where } s' = g^\star(s, a) + \epsilon, \quad \epsilon \sim \mathcal{N}(0, \sigma^2 I), \tag{1}$$

where $g^\star$ is unknown and the level of Gaussian noise σ is known. To learn $g^\star$, we utilize a function class $\mathcal{G} \subset \mathcal{S} \times \mathcal{A} \mapsto \mathcal{S}$. Together with the known Gaussian noise, we have $\mathcal{P} = \{\mathcal{N}(g(s, a), \sigma^2 I) : g \in \mathcal{G}\}$. Extending our results to Gaussian noise with general positive definite covariance matrix is straightfoward; for analysis simplicity, we focus on covariance matrix $\sigma^2 I$.

Examples: One example is the Kernelized Nonlinear Regulator (KNR) model, where $g^\star(s, a) = W^\star\phi(s, a)$ where $\phi : \mathcal{S} \times \mathcal{A} \mapsto \mathcal{H}$ with $\mathcal{H}$ being some Hilbert space (e.g., a Reproducing Kernel Hilbert Space) Kakade et al. (2020b); Mania et al. (2020). Note that this kernelized model captures both linear and nonlinear dynamical system such as hybrid linear systems and has been used in practice extensively (Ko et al., 2007; Deisenroth & Rasmussen, 2011; Fisac et al., 2018; Umlauft et al., 2018). Another example is that $g^\star$ is captured by a Gaussian Process with some pre-defined kernel $k : (\mathcal{S} \times \mathcal{A})^2 \mapsto \mathbb{R}$. Moreover, $\mathcal{G}$ could also be a general function class. For purposes of the theory results, we require these problem settings to satisfy certain regularity conditions, for *e.g.*, bounds on information gain (Srinivas et al., 2009), or, *eluder dimension* (Russo & Roy, 2013) similar to RL literature. In Section 3 and 4, we discuss these examples in detail.

Algorithm 1 `MobILE`: Model-based Imitation Learning and Exploring for ILFO

```
1: Require: IPM class $\mathcal{F}$, dynamics model class $\mathcal{P}$, policy class $\Pi$, bonus function class $\mathcal{B}$.
2: Initialize policy $\pi_0 \in \Pi$, replay buffer $\mathcal{D}_{-1} = \emptyset$.
3: for $t = 0, \cdots, T-1$ do
4:     Execute $\pi_t$ in true environment $P^\star$ to get samples $\tau_t = \{s_k, a_k\}_{k=0}^{H-1} \cup s_H$. Append to replay buffer $\mathcal{D}_t = \mathcal{D}_{t-1} \cup \tau_t$.
5:     Update model and bonus: $\widehat{P}_{t+1} : \mathcal{S} \times \mathcal{A} \to \mathcal{S}$ and $b_{t+1} : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^+$ using buffer $\mathcal{D}_t$.
6:     Get i.i.d expert states $\mathcal{D}_e \equiv \{s_i^e\}_{i=1}^N$.          (for simplicity of theoretical analysis)
7:     Optimistic model-based min-max IL: obtain $\pi_{t+1}$ by solving equation (2) with $\widehat{P}_{t+1}, b_{t+1}, \mathcal{D}_e$.
8: end for
9: Return $\pi_T$.
```

3 Algorithm

We introduce `MobILE`—**Mo**del-**b**ased **I**mitation **L**earning and **E**xploring for ILFO. `MobILE` utilizes (a) a discriminator class $\mathcal{F}$ for performing Integral Probability Metric (IPM) based distribution matching, (b) a dynamics model class $\mathcal{P}$ for model learning, (c) a bonus function class $\mathcal{B}$ for exploration, (d) a policy class Π for policy learning. `MobILE` (in Algorithm 1) iteratively learns a dynamics model $\widehat{P}$ and uses this to learn a policy that aims to match the expert's state visitation using discriminators $\mathcal{F}$. At every iteration, `MobILE` involves:

1. **Dynamics Model Learning:** execute current policy in the environment to obtain state-action-next state (s, a, s') triples which are appended to the buffer $\mathcal{D}$. Fit a dynamics model $\widehat{P}$ on $\mathcal{D}$.
2. **Intrinsic Bonus Design:** design bonus to incentivize exploration where the learnt dynamics model is uncertain, i.e. the bonus $b(s, a)$ is large at state s where $\widehat{P}(\cdot|s, a)$ is uncertain in terms of estimating $P^\star(\cdot|s, a)$, while $b(s, a)$ is small where $\widehat{P}(\cdot|s, a)$ is certain.
3. **Trading off Imitation Against Exploration:** Given discriminators $\mathcal{F}$, a learned model $\widehat{P}$, bonus b and expert dataset $\mathcal{D}_e$, perform *optimistic model-based* optimization of the IPM objective:

$$\pi_{t+1} \leftarrow \arg\min_{\pi\in\Pi} \max_{f\in\mathcal{F}} L(\pi, f; \widehat{P}, b, \mathcal{D}_e) := \mathbb{E}_{(s,a)\sim d^\pi_{\widehat{P}}} \left[f(s) - b(s, a)\right] - \mathbb{E}_{s\sim\mathcal{D}_e} \left[f(s)\right]. \quad (2)$$

Intuitively, the bonus cancels out the power of discriminators in parts of state space where the learned model $\widehat{P}$ is inaccurate, thus allowing `MobILE` to explore. Below we elaborate on the components of `MobILE` while highlighting its key property—which is to trade-off *exploration and imitation*.

3.1 Components of MobILE

This section presents an overview of `MobILE` by instantiating these with the general function approximation class $\mathcal{G}$; refer to the appendix for instantiating `MobILE` in KNR case.

Learning the dynamics model: For model fitting in line 5, given $\mathcal{G}$, and the assumption on the ground truth model (Eq. 1), one can learn $\widehat{g}_t$ via least squares, i.e., $\widehat{g}_t = \arg\min_{g\in\mathcal{G}} \sum_{s,a,s'\in\mathcal{D}_t} \|g(s,a) - s'\|_2^2$, and set $\widehat{P}_t(\cdot|s, a) = \mathcal{N}\left(\widehat{g}_t(s, a), \sigma^2 I\right)$. In practice, there are a variety of developments, for e.g., learning a standard multi-layer perceptron (MLP) based Gaussian Dynamics model (Rajeswaran et al., 2020) that can be used to effectively handle this step.

Intrinsic Bonus Parameterization: We utilize bonuses to incentivize the policy to explore unknown parts of the state space for improved model learning (and better distribution matching as a result). For the general class $\mathcal{G}$, given the least square solution $\widehat{g}_t$, we can define a version space $\mathcal{G}_t$: $\mathcal{G}_t = \left\{g \in \mathcal{G} : \sum_{i=0}^{t-1}\sum_{h=0}^{H-1} \|g(s_h^t, a_h^t) - \widehat{g}_t(s_h^t, a_h^t)\|_2^2 \le z_t\right\}$, with z_t being a hyper parameter. The version space $\mathcal{G}_t$ is an *ensemble of functions* $g \in \mathcal{G}$ with training error on $\mathcal{D}_t$ almost as small as the training error of the least square solution $\widehat{g}_t$. In other words, version space $\mathcal{G}_t$ contains functions that nearly agree on $\mathcal{D}_t$. The uncertainty measure at (s, a) is then the *maximum disagreement* among models in the ensemble $\mathcal{G}_t$, with $b_t(s, a) \propto \sup_{f_1,f_2\in\mathcal{G}_t} \|f_1(s, a) - f_2(s, a)\|_2$. Since functions in $\mathcal{G}_t$ agree on $\mathcal{D}_t$, a large $b_t(s, a)$ indicates that (s, a) is novel. See example 1 for more details.

Empirically, an ensemble's model disagreement (Osband et al., 2018; Azizzadenesheli et al., 2018; Burda et al., 2019; Pathak et al., 2019; Lowrey et al., 2019) has been used for designing bonuses that incentivize exploration. This paper utilizes an ensemble of neural networks to approximate the version space $\mathcal{G}_t$, where each model is trained on $\mathcal{D}_t$ using SGD with different initialization. The bonus is set as a function of maximum disagreement among the ensemble's predictions.

Optimistic model-based min-max IL: For model-based imitation (line 7), `MobILE` takes the current model $\widehat{P}_t$ and discriminators $\mathcal{F}$ as inputs and searches for a policy that approximately minimizes the divergence defined by $\widehat{P}_n$ and $\mathcal{F}$: $d_t(\pi, \pi^e) := \max_{f\in\mathcal{F}} \left[\mathbb{E}_{s,a\sim d^{\pi}_{\widehat{P}_t}} (f(s) - b_t(s,a)) - \mathbb{E}_{s\sim d^{\pi^e}} f(s)\right]$. Note that, for a fixed π, the $\arg\max_{f\in\mathcal{F}}$ is identical with or without the bonus term, since $\mathbb{E}_{s,a\sim d^{\pi}_{\widehat{P}_t}} b_t(s,a)$ is independent of f. Thus the bonus does not affect the search for the most powerful discriminators. In our implementation, we use the Maximum Mean Discrepancy (MMD) with a Radial Basis Function (RBF) kernel to model discriminators $\mathcal{F}$[2]. We compute $\arg\min_\pi d_n(\pi, \pi^e)$ by iteratively (1) computing the $\arg\max$ discriminator f given current π, and (2) using policy gradient (PG) methods (e.g., TRPO) to perform gradient descent on π under $\widehat{P}_t$ with $f(s,a) - b(s,a)$ as the cost. Specifically, for line 7, we iterate between the following two steps; (1) update discriminator: $\hat{f} = \arg\max_{f\in\mathcal{F}} \mathbb{E}_{s\sim d^{\hat{\pi}}_{\widehat{P}_t}} f(s) - \mathbb{E}_{s\sim d^{\pi^e}} f(s)$ and (2) update policy: $\hat{\pi} = \hat{\pi} - \eta \cdot \nabla_\pi V^{\hat{\pi}}_{\widehat{P}_t, \hat{f}-b}$, where the PG step uses the learnt dynamics model $\widehat{P}_t$ and the optimistic IPM cost $\hat{f}(s) - b_t(s,a)$.

3.2 MobILE: Explore And Imitate Dilemma

We note that `MobILE` automatically *trades-off exploration and imitation*. More specifically, the bonus is designed such that it has high values for states in the state space that have not been visited, and low values for states in the state space that have been frequently visited by the sequence of learned policies so far. By incorporating the bonus into the discriminator $f \in \mathcal{F}$ (e.g., $\widetilde{f}(s,a) = f(s) - b_t(s,a)$), we diminish the power of discriminator f at novel state-action space regions. Thus, when matching the learner's states to expert's states, we relax the state-matching constraint at those novel regions so that exploration is encouraged. On the other hand, for well explored state regions, we force the learner's states to match the expert's states using the full power of the discriminators.

Is exploration *necessary* to solve the ILFO problem? Conventional wisdom from the IL literature, where, the learner has access to expert actions suggests that exploration is not required for successful imitation – indeed, standard IL approaches such as BC (Ross & Bagnell, 2010) and DAgger (Ross et al., 2011a) reduce IL to supervised learning and do not rely on exploration to yield successful results. Section 4.2 shows that exploration is *necessary* to solve the ILFO problem efficiently.

4 Analysis

Recall that Algorithm 1 generates one state-action trajectory $\tau^t := \{s^t_h, a^t_h\}_{h=0}^H$ at iteration t and estimates model $\widehat{P}_t$ based on $\mathcal{D}_t$ which contains previous t trajectories $\tau^0, \ldots, \tau^{t-1}$. We provide a unified analysis under the assumption that the model fitting step gives us a calibrated model (Curi et al., 2020), i.e. one that offers predictions coupled with confidence interval. More specifically:

Assumption 2 (Calibrated Model)**.** *For all iteration t with $t \in \mathbb{N}$, with probability $1-\delta$, we have a model $\widehat{P}_t$ and its associated uncertainty measure $\sigma_t : \mathcal{S}\times\mathcal{A} \mapsto \mathbb{R}^+$, such that for all $s,a \in \mathcal{S}\times\mathcal{A}$:*[3] $\left\|\widehat{P}_t(\cdot|s,a) - P^\star(\cdot|s,a)\right\|_1 \leq \min\{\sigma_t(s,a), 2\}$.

Example 1 (General class $\mathcal{G}$)**.** *Assume $\mathcal{G}$ is discrete (but could be exponentially large with complexity measure $\ln(|\mathcal{G}|)$), and $\sup_{g\in\mathcal{G},s,a} \|g(s,a)\|_2 \leq G \in \mathbb{R}^+$. For model learning, $\widehat{g}_t$ is the the least square solution, i.e., $\widehat{g}_t = \arg\min_{g\in\mathcal{G}} \sum_{k=0}^{t-1}\sum_{h=0}^{H-1} \left\|g(s^k_h, a^k_h) - s^k_{h+1}\right\|_2^2$. For the uncertainty measure, compute a version space $\mathcal{G}_t = \left\{g : \sum_{k=0}^{t-1}\sum_{h=0}^{H-1} \left\|g(s^k_h, a^k_h) - \widehat{g}_t(s^k_h, a^k_h)\right\|_2^2 \leq z_t\right\}$, where*

[2]For MMD with kernel k, $\mathcal{F} = \{w^\top\phi(s,a) | \|w\|_2 \leq 1\}$ where, $<\phi(s,a), \phi(s',a')> = k((s,a),(s',a'))$.

[3]the uncertainty measure $\sigma_t(s,a)$ will depend on the input failure probability δ, which we drop here for notational simplicity. When we introduce specific examples, this dependence on δ will be made explicit.

$z_t = 2\sigma^2 G^2 \ln(2t^2|\mathcal{G}|/\delta)$. Set $\sigma_t(s,a) = \frac{1}{\sigma}\max_{g_1,g_2\in\mathcal{G}} \|g_1(s,a) - g_2(s,a)\|_2$, i.e., the maximum disagreement between any two functions in the version space $\mathcal{G}_t$. Refer to Proposition 12 for more details.

The use of maximum disagreement above motivates our practical implementation where we use an ensemble of neural networks to approximate the version space and use the maximum disagreement among the models' predictions as the bonus. We refer readers to Appendix for more details.

4.1 Regret Bound

We bound the regret with the quantity named *Information Gain $\mathcal{I}$* (Srinivas et al., 2009):

$$\mathcal{I}_T := \max_{\text{Alg}} \mathbb{E}_{\text{Alg}}\left[\sum_{t=0}^{T-1}\sum_{h=0}^{H-1} \min\left\{\sigma_t^2(s_h^t, a_h^t), 1\right\}\right], \tag{3}$$

where Alg is any adaptive algorithm (thus including Algorithm 1) that maps from history before iteration t to some policy $\pi_t \in \Pi$. After the main theorem, we give concrete examples for $\mathcal{I}_T$ where we show that $\mathcal{I}_T$ has extremely mild growth rate with respect to T (i.e., logarithimic). Denote V^π as the expected total cost of π under the true cost function c and the real dynamics $P^\star$.

Theorem 3 (Main result)**.** *Assume model learning is calibrated (i.e., Assumption 2 holds for all t). In Algorithm 1, set bonus $b_t(s,a) := H\min\{\sigma_t(s,a), 2\}$. There exists a set of parameters, such that after running Algorithm 1 for T iterations, we have:*

$$\mathbb{E}\left[\min_{t\in[0,\dots,T-1]} V^{\pi_t} - V^{\pi^e}\right] \leq O\left(\frac{H^{2.5}\sqrt{\mathcal{I}_T}}{\sqrt{T}} + H\sqrt{\frac{\ln(TH|\mathcal{F}|)}{N}}\right).$$

Appendix A contains proof of Theorem 3. This theorem indicates that as long as $\mathcal{I}_T$ grows $o(\sqrt{T})$, we find a policy that is at least as good as the expert policy when T and N approach infinity. We present several remarks below regarding specializing the above unified theorem to special instances. For KNR and general $\mathcal{G}$ with bounded Eluder dimension (Osband & Van Roy (2014)), we can show that $\mathcal{I}_T$ grows as $\ln(TH)$. Thus, the algorithm converges to the expert policy at the rate $\widetilde{O}(1/\sqrt{T} + 1/\sqrt{N})$.

Corollary 4 (Bounded Eluder dimension (Example 1))**.** *For general $\mathcal{G}$, we assume that $\mathcal{G}$ has Eluder-dimension $d_E(\epsilon)$ (Definition 3 in Osband & Van Roy (2014)). Denote $d_E = d_E(1/TH)$. We can upper bound the information gain: $\mathcal{I}_T = O\left(Hd_E + d_E\ln(T^3H|\mathcal{G}|)\ln(TH)\right)$ (see 14 for details). Thus, $\mathbb{E}\left[\min_{t\in[0,\dots,T-1]} V^{\pi_t} - V^{\pi^e}\right] = \widetilde{O}\left(\frac{H^3\sqrt{d_E\ln(TH|\mathcal{G}|)}}{\sqrt{T}} + H\sqrt{\frac{\ln(TH|\mathcal{F}|)}{N}}\right)$.*

Thus as long as $\mathcal{G}$ has bounded complexity in terms of the Eluder dimension, the maximum disagreement among models in the version space $\mathcal{G}_t$ leads to near-optimal guarantees.

4.2 Is Exploration Necessary in ILFO?

To answer this question, we present a novel reduction of the ILFO problem to a Multi-Armed Bandit (MAB) problem, for which we know exploration is *necessary* (Bubeck & Cesa-Bianchi, 2012); this indicates that exploration is *necessary* to solve the ILFO problem efficiently.

Consider a MAB problem with A actions $\{a_i\}_{i=1}^A$. Each action's ground truth reward $r_i \sim \mathcal{N}(\mu_i, 1)$ is from a Gaussian with mean μ_i and variance 1. Without loss of generality, assume a_1 is the optimal arm, i.e., $\mu_1 > \mu_i \ \forall\, i \neq 1$. We convert this MAB instance into an MDP. Specifically, set $H = 2$. Suppose we have a fixed initial state s_0 which has A actions. For the one step transition, we have $P(\cdot|s_0, a_i) = \mathcal{N}(\mu_i, 1)$, i.e., $g^*(s_0, a_i) = \mu_i$. Here we denote the optimal expert policy π^e as $\pi^e(s_0) = a_1$, i.e., expert policy picks the optimal arm in the MAB instance. Hence, when executing π^e, we note that the state s_1 generated from π^e is simply the stochastic reward of a_1 in the original MAB instance. Assume that we have observed infinitely many such s_1 from the expert policy π^e, i.e., we have infinitely many samples of expert state data, i.e., $N \to \infty$. Note, however, we do not have access to expert actions (since this is the ILFO setting). This expert data is equivalent to revealing the optimal arm's mean reward μ_1 to the MAB learner a priori. Hence solving the ILFO problem on this MDP is no easier than solving the original MAB instance with one additional piece of information which is that optimal arm's mean reward is μ_1 (but the identity of the best arm is unknown).

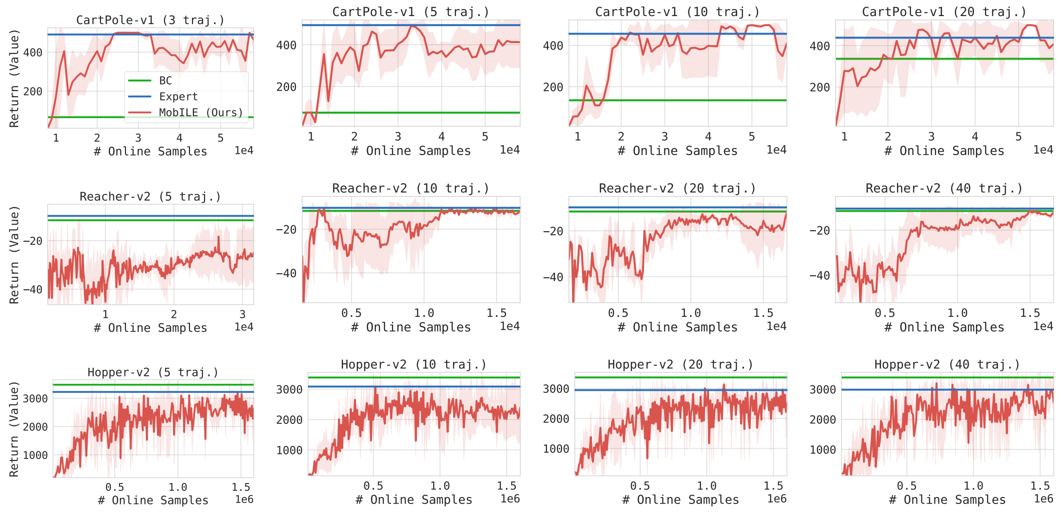

Figure 1: Comparing MobILE (red) against BC (green) and Expert (blue) on Cartpole-v1 (1st row), Reacher-v2 (2nd row), Hopper-v2 (3th row) by varying amounts of expert trajectories. MobILE matches or exceeds BC's performance despite BC having access to expert actions.

Theorem 5. *Consider solving a Gaussian MAB with the additional information that the optimal arm's mean reward is μ (i.e., μ is known but the identity of the best arm is unknown). For any algorithm, there exists a MAB instance with number of arms $A \geq 2$, such that the expected regret is still $\Omega(\sqrt{AT})$, i.e., the additional information does not help improving the worst-case regret bound.*

Appendix A contains proof of Theorem 5. Theorem 5 shows that solving ILFO, with even infinite expert data, is at least as hard as solving the MAB problem with the known optimal arm's mean reward which incurs the same worst case $\sqrt{AT}$ regret bound as the one in the classic MAB setting. In contrast, in traditional IL and full-information supervised learning settings, methods like BC have sample complexities that scale as poly $\ln(A)$. The necessity for exploration in ILFO is manifested in the *exponential gap* in the sample complexity dependence on A between IL and the IFLO setting.

5 Experiments

This section seeks to answer the following questions:

- **MobILE's behavior:** How does MobILE perform relative to other model-based ILFO methods?
- **Importance of optimism:** Is optimism important for solving ILFO with MobILE? How does MobILE behave with varying levels of optimism?
- **Stochastic environments:** How does MobILE perform in MDPs with stochastic dynamics?

We consider tasks from Open AI Gym (Brockman et al., 2016) simulated with Mujoco (Todorov et al., 2012) using Cartpole-v1, Reacher-v2, and Hopper-v2 tasks. For Reacher-v2, following Sun et al. (2019b), we discretize every action dimension into five equally spaced bins. For Hopper-v2, we work with continuous actions. We use TRPO (Schulman et al., 2015) for training an expert with value $\approx 460, -10, 3000$ for Cartpole-v1, Reacher-v2, Hopper-v2. We setup Hopper-v2 similar to prior model-based RL works (Kurutach et al., 2018; Luo et al., 2018). We report results with $3, 5, 10, 20$ expert trajectories for Cartpole-v1 and with $5, 10, 20, 40$ trajectories for Reacher-v2 and Hopper-v2; all of our results are averaged over 3 seeds.

We benchmark MobILE against BC instead of other model-based ILFO algorithms. Note, (a) BC upperbounds performance of other model based approaches such as BC-O (Torabi et al., 2018), (b) other approaches like Edwards et al. (2019) require deterministic environments and discrete actions, whereas, MobILE works with stochastic environments and continuous actions. Unlike MobILE and other ILFO methods, BC has access to expert actions and thus is a non-trivial benchmark. We report the average across 3 seeds of the best policy obtained with BC. See Appendix C for more details.

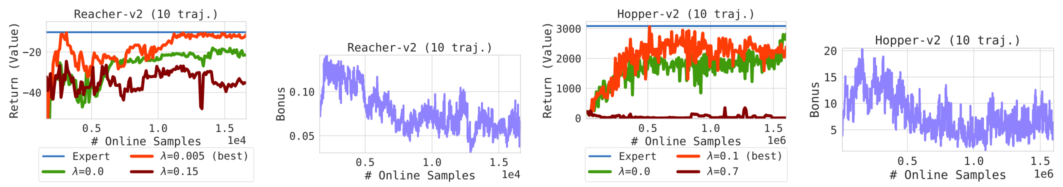

Figure 2: From left to right: 1st/3rd plot – learning curves (without error bars to avoid clutter) with varying λ for `Reacher-v2`, `Hopper-v2`; 2nd/4th plot — bonus variation vs. algorithm progress for optimal λ for `Reacher-v2`, `Hopper-v2`. Higher λ implies larger bonuses added to the rewards. Note that lower λ leads to sample inefficient learning; higher λ also leads to highly sub-optimal behavior. Successful imitation requires trading off exploration and imitation with intermediate λ.

5.1 BENCHMARKING `MobILE` ON MUJOCO SUITE

Figure 1 compares `MobILE` with BC. Note that `MobILE` matches or exceeds BC's performance despite BC having access to expert actions while `MobILE` functions *without* expert actions. This implies `MobILE` improves upon BC-O (Torabi et al., 2018), since it is outperformed by BC. Note that BC offers strong results in these benchmarks owing to deterministic transition dynamics; in section 5.3, we compare `MobILE` with BC on environments that exhibit stochastic transition dynamics.

5.2 IMPORTANCE OF THE OPTIMISTIC MDP CONSTRUCTION

We consider `Reacher-v2` and `Hopper-v2` with 10 expert trajectories. We vary the level of exploration performed by `MobILE` by varying λ (the hyper-parameter that trades off discriminator cost against the bonus – for details, see Appendix section C.1). Figure 2 indicates that the value of λ tends to influence `MobILE`'s performance. In particular, $\lambda = 0$ implies the algorithm is not *explicitly* incentivized to explore; it explores because of sampling actions from a stochastic policy. We observe that lower λ's are associated with sample inefficiency in terms of number of online interactions needed to solve the problem. A large λ however implies the algorithm over-explores and is not adequately rewarded for distribution matching. The key to the success of `MobILE` is balancing exploration with imitation. Empirically, we observe an initial increase in the bonus signifying that the algorithm explores followed by a decay as the algorithm trades-off exploration for imitation.

5.3 PERFORMANCE WITH STOCHASTIC ENVIRONMENTS

We consider a stochastic variant of `Cartpole-v0`, wherein, zero mean additive Gaussian noise is added to the transition dynamics. The variance of the noise is not known to the learner. Figure 3 presents the results of BC in comparison to `MobILE` for this problem. We observe that this minor modification to the environment leads to rapid degradation of BC's performance. This result indicates that stochastic transition dynamics are not straightforward for algorithms relying on BC, for e.g. BC-O, to reliably solve while `MobILE` successfully solves the ILFO problem even with stochastic transition dynamics.

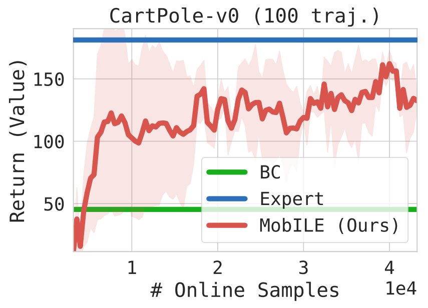

Figure 3: Comparing `MobILE` against BC in a stochastic variant of `Cartpole-v0`. While BC fails with stochastic transition dynamics, `MobILE` performs reliable imitation.

6 CONCLUSIONS

We introduce `MobILE`, a model-based ILFO approach that works for MDPs with stochastic transition dynamics and continuous action spaces. `MobILE` uses ideas from self-supervision by learning dynamics models and using this for bonus-based optimistic imitation. `MobILE` automatically balances exploration and imitation and provably matches the expert's behavior. We show that exploration is *necessary* in ILFO as it is fundamentally harder than IL where the experts' actions are known. Our experiments on benchmark tasks (with discrete and continuous actions, stochastic and deterministic transitions) demonstrate that `MobILE` matches the expert efficiently and reliably.

REFERENCES

Pieter Abbeel and Andrew Y. Ng. Apprenticeship learning via inverse reinforcement learning. In *ICML*. ACM, 2004.

Peter Auer, Nicolo Cesa-Bianchi, and Paul Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine learning*, 47(2):235–256, 2002.

Yusuf Aytar, Tobias Pfaff, David Budden, Tom Le Paine, Ziyu Wang, and Nando de Freitas. Playing hard exploration games by watching youtube. In *NeurIPS*, pp. 2935–2945, 2018.

Mohammad Gheshlaghi Azar, Ian Osband, and Rémi Munos. Minimax regret bounds for reinforcement learning. In *International Conference on Machine Learning*, pp. 263–272, 2017.

Kamyar Azizzadenesheli, Emma Brunskill, and Animashree Anandkumar. Efficient exploration through bayesian deep q-networks. In *ITA*, pp. 1–9. IEEE, 2018.

Ronen I. Brafman and Moshe Tennenholtz. R-max - a general polynomial time algorithm for near-optimal reinforcement learning. *J. Mach. Learn. Res.*, 3:213–231, 2001.

Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. Openai Gym. *arXiv preprint arXiv:1606.01540*, 2016.

Sébastien Bubeck and Nicolò Cesa-Bianchi. Regret analysis of stochastic and non-stochastic multi-armed bandit problems. *Found. Trends Mach. Learn*, 5(1):1–122, 2012.

Yuri Burda, Harrison Edwards, Amos Storkey, and Oleg Klimov. Exploration by random network distillation. *arXiv preprint arXiv:1810.12894*, 2018.

Yuri Burda, Harrison Edwards, Amos J. Storkey, and Oleg Klimov. Exploration by random network distillation. In *ICLR*. OpenReview.net, 2019.

Sebastian Curi, Felix Berkenkamp, and Andreas Krause. Efficient model-based reinforcement learning through optimistic policy search and planning. *arXiv preprint arXiv:2006.08684*, 2020.

Hal Daumé, John Langford, and Daniel Marcu. Search-based structured prediction. *Mach. Learn.*, 75(3):297–325, June 2009. ISSN 0885-6125. doi: 10.1007/s10994-009-5106-x. URL `https://doi.org/10.1007/s10994-009-5106-x`.

Marc Deisenroth and Carl E. Rasmussen. PILCO: A model-based and data-efficient approach to policy search. In *International Conference on Machine Learning*, pp. 465–472, 2011.

Siddharth Desai, Ishan Durugkar, Haresh Karnan, Garrett Warnell, Josiah Hanna, Peter Stone, and AI Sony. An imitation from observation approach to transfer learning with dynamics mismatch. *Advances in Neural Information Processing Systems*, 33, 2020.

Prafulla Dhariwal, Christopher Hesse, Oleg Klimov, Alex Nichol, Matthias Plappert, Alec Radford, John Schulman, Szymon Sidor, Yuhuai Wu, and Peter Zhokhov. Openai baselines. `https://github.com/openai/baselines`, 2017.

Ashley D. Edwards, Himanshu Sahni, Yannick Schroecker, and Charles L. Isbell Jr. Imitating latent policies from observation. In *ICML*, 2019.

Chelsea Finn, Sergey Levine, and Pieter Abbeel. Guided cost learning: Deep inverse optimal control via policy optimization. In *ICML*, 2016.

Jaime F Fisac, Anayo K Akametalu, Melanie N Zeilinger, Shahab Kaynama, Jeremy Gillula, and Claire J Tomlin. A general safety framework for learning-based control in uncertain robotic systems. *IEEE Transactions on Automatic Control*, 64(7):2737–2752, 2018.

Seyed Kamyar Seyed Ghasemipour, Richard Zemel, and Shixiang Gu. A divergence minimization perspective on imitation learning methods. In *Conference on Robot Learning*, pp. 1259–1277. PMLR, 2020.

Shixiang Gu, Timothy Lillicrap, Ilya Sutskever, and Sergey Levine. Continuous deep q-learning with model-based acceleration, 2016.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. *CoRR*, abs/1801.01290, 2018.

Jonathan Ho and Stefano Ermon. Generative adversarial imitation learning. *CoRR*, abs/1606.03476, 2016.

Michael Janner, Justin Fu, Marvin Zhang, and Sergey Levine. When to trust your model: Model-based policy optimization. *CoRR*, abs/1906.08253, 2019.

Sham Kakade, Akshay Krishnamurthy, Kendall Lowrey, Motoya Ohnishi, and Wen Sun. Information theoretic regret bounds for online nonlinear control. *arXiv preprint arXiv:2006.12466*, 2020a.

Sham M. Kakade. A natural policy gradient. In *NIPS*, pp. 1531–1538, 2001.

Sham M. Kakade, Michael J. Kearns, and John Langford. Exploration in metric state spaces. In *ICML*, 2003.

Sham M. Kakade, Akshay Krishnamurthy, Kendall Lowrey, Motoya Ohnishi, and Wen Sun. Information theoretic regret bounds for online nonlinear control. In *NeurIPS*, 2020b.

Liyiming Ke, Matt Barnes, Wen Sun, Gilwoo Lee, Sanjiban Choudhury, and Siddhartha Srinivasa. Imitation learning as f-divergence minimization. *arXiv preprint arXiv:1905.12888*, 2019.

Michael Kearns and Satinder Singh. Near optimal reinforcement learning in polynomial time. *Machine Learning*, 49(2-3):209–232, 2002a.

Michael Kearns and Satinder Singh. Near-optimal reinforcement learning in polynomial time. *Machine learning*, 49(2-3):209–232, 2002b.

Jonathan Ko, Daniel J Klein, Dieter Fox, and Dirk Haehnel. Gaussian processes and reinforcement learning for identification and control of an autonomous blimp. In *Proceedings 2007 ieee international conference on robotics and automation*, pp. 742–747. IEEE, 2007.

Thanard Kurutach, Ignasi Clavera, Yan Duan, Aviv Tamar, and Pieter Abbeel. Model-ensemble trust-region policy optimization. In *ICLR*. OpenReview.net, 2018.

Thomas Lampe and Martin A. Riedmiller. Approximate model-assisted neural fitted q-iteration. In *IJCNN*, pp. 2698–2704. IEEE, 2014.

Tor Lattimore and Csaba Szepesvári. *Bandit Algorithms*. Cambridge University Press, 2020. doi: 10.1017/9781108571401.

Sergey Levine, Chelsea Finn, Trevor Darrell, and Pieter Abbeel. End-to-end training of deep visuomotor policies. *CoRR*, abs/1504.00702, 2015.

Weiwei Li and Emanuel Todorov. Iterative linear quadratic regulator design for nonlinear biological movement systems. In *ICINCO*, pp. 222–229, 2004.

Kendall Lowrey, Aravind Rajeswaran, Sham Kakade, Emanuel Todorov, and Igor Mordatch. Plan Online, Learn Offline: Efficient Learning and Exploration via Model-Based Control. In *International Conference on Learning Representations (ICLR)*, 2019.

Yuping Luo, Huazhe Xu, Yuanzhi Li, Yuandong Tian, Trevor Darrell, and Tengyu Ma. Algorithmic framework for model-based deep reinforcement learning with theoretical guarantees. *arXiv preprint arXiv:1807.03858*, 2018.

Horia Mania, Michael I Jordan, and Benjamin Recht. Active learning for nonlinear system identification with guarantees. *arXiv preprint arXiv:2006.10277*, 2020.

Charith Mendis, Cambridge Yang, Yewen Pu, Saman P. Amarasinghe, and Michael Carbin. Compiler auto-vectorization with imitation learning. In *NeurIPS*, pp. 14598–14609, 2019.

Anusha Nagabandi, Gregory Kahn, Ronald S. Fearing, and Sergey Levine. Neural network dynamics for model-based deep reinforcement learning with model-free fine-tuning. In *IEEE International Conference on Robotics and Automation*, pp. 7559–7566, 2018.

Andrew Y. Ng and Stuart Russell. Algorithms for inverse reinforcement learning. In *Proc. ICML*, pp. 663–670, 2000.

Ian Osband and Benjamin Van Roy. Model-based reinforcement learning and the Eluder dimension. In *Advances in Neural Information Processing Systems*, pp. 1466–1474, 2014.

Ian Osband, John Aslanides, and Albin Cassirer. Randomized prior functions for deep reinforcement learning. *CoRR*, abs/1806.03335, 2018.

Deepak Pathak, Dhiraj Gandhi, and Abhinav Gupta. Self-supervised exploration via disagreement. In *ICML*, pp. 5062–5071, 2019.

Xue Bin Peng, Pieter Abbeel, Sergey Levine, and Michiel van de Panne. Deepmimic: example-guided deep reinforcement learning of physics-based character skills. *ACM Trans. Graphics*, 2018.

D. A. Pomerleau. Alvinn: An autonomous land vehicle in a neural network. Technical report, CMU, 1989.

Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. In *Advances in Neural Information Processing Systems*, pp. 1177–1184, 2008.

Aravind Rajeswaran, Kendall Lowrey, Emanuel Todorov, and Sham Kakade. Towards Generalization and Simplicity in Continuous Control. In *NIPS*, 2017.

Aravind Rajeswaran, Igor Mordatch, and Vikash Kumar. A game theoretic framework for model based reinforcement learning. *ArXiv*, abs/2004.07804, 2020.

Stéphane Ross and Drew Bagnell. Efficient reductions for imitation learning. In Yee Whye Teh and D. Mike Titterington (eds.), *AISTATS*, JMLR Proceedings, pp. 661–668, 2010.

Stephane Ross and J Andrew Bagnell. Reinforcement and imitation learning via interactive no-regret learning. *arXiv preprint arXiv:1406.5979*, 2014.

Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pp. 627–635, 2011a.

Stéphane Ross, Geoffrey J. Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *AISTATS*, pp. 627–635, 2011b.

Daniel Russo and Benjamin Van Roy. Eluder dimension and the sample complexity of optimistic exploration. In *NIPS*, pp. 2256–2264, 2013.

Daniel Russo and Benjamin Van Roy. Learning to optimize via posterior sampling. *Mathematics of Operations Research*, 39(4):1221–1243, 2014.

Karl Schmeckpeper, Oleh Rybkin, Kostas Daniilidis, Sergey Levine, and Chelsea Finn. Reinforcement learning with videos: Combining offline observations with interaction. *CoRR*, abs/2011.06507, 2020.

John Schulman, Sergey Levine, Philipp Moritz, Michael I. Jordan, and Pieter Abbeel. Trust region policy optimization. *CoRR*, abs/1502.05477, 2015.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *CoRR*, abs/1707.06347, 2017.

Yuda Song, Aditi Mavalankar, Wen Sun, and Sicun Gao. Provably efficient model-based policy adaptation. In *International Conference on Machine Learning*, pp. 9088–9098. PMLR, 2020.

Niranjan Srinivas, Andreas Krause, Sham M. Kakade, and Matthias Seeger. Gaussian process optimization in the bandit setting: No regret and experimental design. *arXiv preprint arXiv:0912.3995*, 2009.

Wen Sun, Arun Venkatraman, Geoffrey J Gordon, Byron Boots, and J Andrew Bagnell. Deeply aggrevated: Differentiable imitation learning for sequential prediction. *arXiv preprint arXiv:1703.01030*, 2017.

Wen Sun, Nan Jiang, Akshay Krishnamurthy, Alekh Agarwal, and John Langford. Model-based rl in contextual decision processes: Pac bounds and exponential improvements over model-free approaches. In *Conference on Learning Theory*, pp. 2898–2933. PMLR, 2019a.

Wen Sun, Anirudh Vemula, Byron Boots, and Drew Bagnell. Provably efficient imitation learning from observation alone. In *ICML*, volume 97. PMLR, 2019b.

Ilya Sutskever, James Martens, George E. Dahl, and Geoffrey E. Hinton. On the importance of initialization and momentum in deep learning. In *ICML*, volume 28, 2013.

R. S. Sutton. First results with dyna, an integrated architecture for learning, planning, and reacting. In *Neural Networks for Control*, pp. 179–189. The MIT Press: Cambridge, MA, USA, 1990.

Emanuel Todorov, Tom Erez, and Yuval Tassa. MuJoCo: A physics engine for model-based control. In *IEEE International Conference on Intelligent Robots and Systems*, pp. 5026–5033, 2012.

Faraz Torabi, Garrett Warnell, and Peter Stone. Behavioral cloning from observation. In *IJCAI*, pp. 4950–4957, 2018.

Jonas Umlauft, Lukas Pöhler, and Sandra Hirche. An uncertainty-based control lyapunov approach for control-affine systems modeled by gaussian process. *IEEE Control Systems Letters*, 2(3): 483–488, 2018.

Chao Yang, Xiaojian Ma, Wenbing Huang, Fuchun Sun, Huaping Liu, Junzhou Huang, and Chuang Gan. Imitation learning from observations by minimizing inverse dynamics disagreement. In *NeurIPS*, 2019.

Brian D Ziebart, Andrew L Maas, J Andrew Bagnell, and Anind K Dey. Maximum entropy inverse reinforcement learning. In *Aaai*, volume 8, pp. 1433–1438. Chicago, IL, USA, 2008.

A Analysis of Algorithm 1

We start by presenting the proof for the unified main result in Theorem 3. We then discuss the bounds for special instances individually.

The following lemma shows that under Assumption 2, with $b_t(s,a) = H\min\{\sigma_t(s,a), 2\}$, we achieve *optimism* at all iterations.

Lemma 6 (Optimism). *Assume Assumption 2 holds, and set* $b_t(s,a) = H\min\{\sigma_t(s,a), 2\}$. *For all state-wise cost function* $f : \mathcal{S} \mapsto [0,1]$, *denote the bonus enhance cost as* $\widetilde{f}_t(s,a) := f(s) - b_t(s,a)$. *For all policy* π, *we have the following optimism:*

$$V^{\pi}_{\widehat{P}_t, \widetilde{f}_t} \leq V^{\pi}_{P,f}, \forall t.$$

Proof. In the proof, we drop subscript t for notation simplicity. We consider a fixed function f and policy π. Also let us denote $\widehat{V}^{\pi}$ as the value function of π under $(\widehat{P}, \widetilde{f})$, and V^{π} as the value function under (P, f).

Let us start from $h = H$, where we have $\widehat{V}^\pi_H(s) = V^\pi_H(s) = 0$. Assume inductive hypothesis holds at $h+1$, i.e., for any s, a, we have $\widehat{Q}^\pi_{h+1}(s,a) \leq Q^\pi_{h+1}(s,a)$. Now let us move to h. We have:

$$\begin{aligned}\widehat{Q}^\pi_h(s,a) - Q^\pi_h(s,a) &= \widetilde{f}(s,a) + \mathbb{E}_{s'\sim\widehat{P}(\cdot|s,a)}\widehat{V}^\pi_{h+1}(s') - f(s) - \mathbb{E}_{s'\sim P(\cdot|s,a)}V^\pi_{h+1}(s') \\ &\leq -H\min\{\sigma(s,a),2\} + \mathbb{E}_{s'\sim\widehat{P}(\cdot|s,a)}V^\pi_{h+1}(s') - \mathbb{E}_{s'\sim P(\cdot|s,a)}V^\pi_{h+1}(s') \\ &\leq -H\min\{\sigma(s,a),2\} + H\left\|\widehat{P}(\cdot|s,a) - P(\cdot|s,a)\right\|_1 \\ &\leq -H\min\{\sigma(s,a),2\} + H\min\{\sigma(s,a),2\} = 0,\end{aligned}$$

where the first inequality uses the inductive hypothesis at time step $h+1$. Finally, note that $V^\pi_h(s) = \mathbb{E}_{a\sim\pi(s)}Q^\pi_h(s,a)$, which leads to $\widehat{V}^\pi_h(s) \leq V^\pi_h(s)$. This concludes the induction step. □

The next lemma concerns the statistical error from finite sample estimation of $\mathbb{E}_{s\sim d^{\pi^e}}f(s)$.

Lemma 7. *Fix $\delta \in (0,1)$. For all t, we have that with probability at least $1-\delta$,*

$$\left|\mathbb{E}_{s\sim d^{\pi^e}}f(s) - \sum_{i=1}^N f(s^e_i)/N\right| \leq 2\sqrt{\frac{\ln(2t^2|\mathcal{F}|/\delta)}{N}}, \forall f\in\mathcal{F}.$$

Proof. For any t, we set the failure probability to be $6\delta/(t^2\pi^2)$ at iteration t where we abuse notation and point out that $\pi = 3.14159...$. Thus the total failure probability for all $t \in \mathbb{N}$ is at most δ. We then apply classic Hoeffding inequality to bound $\mathbb{E}_{s\sim d^{\pi^e}}f(s) - \sum_{i=1}^N f(s^e_i)/N$ with the fact that $f(s) \in [0,1]$ for all s. We conclude the proof by taking a union bound over all $f \in \mathcal{F}$. □

Now we conclude the proof for Theorem 3.

Proof of Theorem 3. Assume that Assumption 2 and the event in Lemma 7 hold. Denote the joint of these two events as $\mathcal{E}$. Note that the probability of $\overline{\mathcal{E}}$ is at most 2δ. For notation simplicity, denote $\epsilon_{stats} = 2\sqrt{\frac{\ln(2T^2|\mathcal{F}|/\delta)}{N}}$.

In each model-based planning phase, recall that we perform model-based optimization on the following objective:

$$\pi_t = \arg\min_{\pi\in\Pi}\max_{f\in F}\left[\mathbb{E}_{s,a\sim d^\pi_{\widehat{P}_t}}\left[f(s) - b_t(s,a)\right] - \sum_{i=1}^N f(s^e_i)/N\right].$$

Note that for any π, using the inequality in Lemma 7, we have:

$$\begin{aligned}&\max_{f\in\mathcal{F}_t}\left[\mathbb{E}_{s,a\sim d^\pi_{\widehat{P}_t}}(f(s) - b_t(s,a)) - \sum_{i=1}^N f(s^e_i)/N\right] \\ &= \max_{f\in\mathcal{F}}\left[\mathbb{E}_{s,a\sim d^\pi_{\widehat{P}_t}}(f(s) - b_t(s,a)) - \mathbb{E}_{s\sim d^{\pi^e}}f(s) + \mathbb{E}_{s\sim d^{\pi^e}}f(s) - \sum_{i=1}^N f(s^e_i)/N\right] \\ &\leq \max_{f\in\mathcal{F}}\left[\mathbb{E}_{s,a\sim d^\pi_{\widehat{P}_t}}(f(s) - b_t(s,a)) - \mathbb{E}_{s\sim d^{\pi^e}}f(s)\right] + \max_{f\in F}\left[\mathbb{E}_{s\sim d^{\pi^e}}f(s) - \sum_{i=1}^N f(s^e_i)/N\right] \\ &\leq \max_{f\in\mathcal{F}}\left[\mathbb{E}_{s,a\sim d^\pi_{\widehat{P}_t}}\left(f(s) - b_t(s,a)\right) - \mathbb{E}_{s,a\sim d^{\pi^e}_{\widehat{P}_t}}\left(f(s) - b_t(s,a)\right)\right] + \epsilon_{stats}\end{aligned}$$

where in the last inequality we use optimism from Lemma 6, i.e., $\mathbb{E}_{s,a\sim d^{\pi^e}_{\widehat{P}_t}}(f(s) - b_t(s,a)) \leq \mathbb{E}_{s\sim d^{\pi^e}}f(s)$.

Hence, for π_t, since it is the minimizer and $\pi^e \in \Pi$, we must have:

$$\begin{aligned}
&\max_{f\in\mathcal{F}} \left[\mathbb{E}_{s,a\sim d^{\pi_t}_{\widehat{P}_t}} \left(f(s) - b_t(s,a)\right) - \sum_{i=1}^{N} f(s_i^e)/N \right] \\
&\leq \max_{f\in\mathcal{F}} \left[\mathbb{E}_{s,a\sim d^{\pi^e}_{\widehat{P}_t}} (f(s) - b_t(s,a)) - \sum_{i=1}^{N} f(s_i^e)/N \right] \\
&\leq \max_{f\in\mathcal{F}} \left[\mathbb{E}_{s,a\sim d^{\pi^e}_{\widehat{P}_t}} (f(s) - b_t(s,a)) - \mathbb{E}_{s,a\sim d^{\pi^e}_{\widehat{P}_t}} (f(s) - b_t(s,a)) \right] + \epsilon_{stats} = \epsilon_{stats}.
\end{aligned}$$

Note that $\mathcal{F}$ contains c, we must have:

$$\mathbb{E}_{s,a\sim d^{\pi_t}_{\widehat{P}_t}} \left[c(s) - b_t(s,a)\right] \leq \sum_{i=1}^{N} c(s_i^e)/N + \epsilon_{stats} \leq \mathbb{E}_{s\sim d^{\pi^e}} c(s) + 2\epsilon_{stats},$$

which means that $V^{\pi_t}_{\widehat{P}_t;\widetilde{c}_t} \leq V^{\pi^e} + 2H\epsilon_{stats}$.

Now we compute the regret in episode t. First recall that $b_t(s,a) = H\min\{\sigma_t(s,a), 2\}$, which means that $\|b_t\|_\infty \leq 2H$ as $\|c\|_\infty \leq 1$, which means that $\|c - b_t\|_\infty \leq 2H$. Thus, $\left\|V^{\pi}_{\widehat{P};c-b_t}\right\|_\infty \leq 2H^2$. Recall simulation lemma (Lemma 15), we have:

$$\begin{aligned}
V^{\pi_t} - V^{\pi^e} &\leq V^{\pi_t} - V^{\pi_t}_{\widehat{P}_t;\widetilde{c}_t} + 2H\epsilon_{stats} \\
&= H\mathbb{E}_{s,a\sim d^{\pi_t}} \left[|\widetilde{c}_t(s,a) - c(s)| + 2H^2 \left\|\widehat{P}_t(\cdot|s,a) - P^\star(\cdot|s,a)\right\|_1 \right] + 2H\epsilon_{stat} \\
&= H\mathbb{E}_{s,a\sim d^{\pi_t}} \left[H\min\{\sigma_t(s,a),2\} + 2H^2 \left\|\widehat{P}_t(\cdot|s,a) - P^\star(\cdot|s,a)\right\|_1 \right] + 2H\epsilon_{stat} \\
&\leq H\mathbb{E}_{s,a\sim d^{\pi_t}} \left[H\min\{\sigma_t(s,a),2\} + 2H^2 \min\{\sigma_t(s,a),2\} \right] + 2H\epsilon_{stat} \\
&\leq 3H^3\mathbb{E}_{s,a\sim d^{\pi_t}} \min\{\sigma_t(s,a),2\} + 2H\epsilon_{stat} \\
&\leq 6H^3\mathbb{E}_{s,a\sim d^{\pi_t}} \min\{\sigma_t(s,a),1\} + 2H\epsilon_{stat}
\end{aligned}$$

Now sum over t, and denote $\mathbb{E}_{\pi_t}$ as the conditional expectation conditioned on the history from iteration 0 to $t-1$, we get:

$$\begin{aligned}
\sum_{t=0}^{T-1} \left[V^{\pi_t} - V^{\pi^e} \right] &\leq 6H^2 \sum_{t=0}^{T-1} \mathbb{E}_{\pi_t} \left[\sum_{h=0}^{H-1} \min\{\sigma_t(s_h^t, a_h^t), 1\} \right] + 2HT\epsilon_{stat} \\
&\leq 6H^2 \sum_{t=0}^{T-1} \left[\sqrt{H} \sqrt{ \mathbb{E}_{\pi_t} \sum_{h=0}^{H-1} \min\{\sigma_t^2(s_h^t, a_h^t), 1\} } \right] + 2HT\epsilon_{stat},
\end{aligned}$$

where in the last inequality we use $\mathbb{E}[a^\top b] \leq \sqrt{\mathbb{E}[\|a\|_2^2]\mathbb{E}[\|b\|_2^2]}$.

Recall that π_t are random quantities, add expectation on both sides of the above inequality, and consider the case where $\mathcal{E}$ holds and $\overline{\mathcal{E}}$ holds, we have:

$$\begin{aligned}
\mathbb{E}\left[\sum_{t=0}^{T-1} \left(V^{\pi_t} - V^{\pi^e} \right) \right] &\leq 6H^{2.5}\mathbb{E}\left[\sum_{t=0}^{T-1} \sqrt{ \mathbb{E}_{\pi_t} \sum_{h=0}^{H-1} \min\left\{\sigma_t^2(s_h^t, a_h^t), 1\right\} } \right] + 2HT\epsilon_{stat} + \mathbb{P}(\overline{\mathcal{E}})TH \\
&\leq 6H^{2.5} \left[\sqrt{T} \sqrt{ \mathbb{E}\left[\sum_{t=0}^{T-1}\sum_{h=0}^{H-1} \min\left\{\sigma_t^2(s_h^t, a_h^t), 1\right\} \right] } \right] + 2HT\epsilon_{stat} + 2\delta TH,
\end{aligned}$$

where in the last inequality, we use $\mathbb{E}[a^\top b] \leq \sqrt{\mathbb{E}[\|a\|_2^2]\mathbb{E}[\|b\|_2^2]}$. This implies that that:

$$\mathbb{E}\left[\min_t V^{\pi_t} - V^{\pi^e} \right] \leq \frac{6H^{2.5}}{\sqrt{T}} \sqrt{ \max_{\text{Alg}} \mathbb{E}_{\text{Alg}} \left[\sum_{t=0}^{T-1}\sum_{h=0}^{H-1} \min\left\{\sigma_t^2(s_h^t, a_h^t), 1\right\} \right] } + 2H\epsilon_{stats} + 2H\delta.$$

Set $\delta = 1/(HT)$, we get:

$$\mathbb{E}\left[V^{\pi} - V^{\pi^e}\right] \leq \frac{6H^{2.5}}{\sqrt{T}}\sqrt{\max_{\text{Alg}} \mathbb{E}_{\text{Alg}}\left[\sum_{t=0}^{T-1}\sum_{h=0}^{H-1} \min\left\{\sigma_t^2(s_h^t, a_h^t), 1\right\}\right]} + 2H\sqrt{\frac{\ln(T^3 H|\mathcal{F}|)}{N}} + \frac{2}{T}$$

where Alg is any adaptive mapping that maps from history from $t = 0$ to the end of the $t-1$ iteration to to some policy π_t. This concludes the proof. □

Below we discuss special cases.

A.1 KNRs

Example 2 (KNRs)**.** *We have $g^\star(s,a) = W^\star\phi(s,a)$; $\phi(s,a) \in \mathcal{H}$, and $\sup_{s,a}\|\phi(s,a)\| \leq 1$. We learn $\widehat{P}_t$ via Kernel regression, i.e., $\widehat{g}_t(s,a) = \widehat{W}_t\phi(s,a)$, where,*

$$\widehat{W}_t = \arg\min_{W} \sum_{s,a,s' \in \mathcal{D}_t} \|W\phi(s,a) - s'\|_2^2 + \lambda \|W\|_F^2$$

, where $\|\cdot\|_F$ is the Frobenius norm. The uncertainty measure $\sigma_t(s,a)$ is set as $\sigma_t(s,a) = \frac{\beta_t}{\sigma}\|\phi(s,a)\|_{\Sigma_t^{-1}}$, where,

$$\beta_t = \left\{2\lambda\|W^\star\|_2^2 + 8\sigma^2 \cdot \left[d_s \ln(5) + 2\ln(t^2/\delta) + \ln(4) + \ln\left(\det(\Sigma_t)/\det(\lambda I)\right)\right]\right\}^{1/2},$$

and,

$$\Sigma_t = \sum_{k=0}^{t-1}\sum_{h=1}^{H-1} \phi(s_h^k, a_h^k)\phi(s_h^k, a_h^k)^\top + \lambda I,$$

where $\lambda \in \mathbb{R}^+$.Refer to Proposition 10 for more details.

Remark 8. *Similar to RKHS, Gaussian processes (GPs) offer a calibrated model. Since GPs offer similar regret bounds as RKHS, we refer readers to Curi et al. (2020) for details.*

Corollary 9 (KNRs (Example 2))**.** *For simplicity, consider the finite dimension setting where $\phi : \mathcal{S} \times \mathcal{A} \mapsto \mathbb{R}^d$. We can show that $\mathcal{I}_T = \widetilde{O}\left(Hd + Hdd_s + Hd^2\right)$ (see Proposition 11 for details), where d is the dimension of the feature $\phi(s,a)$. Thus, we have*

$$\mathbb{E}\left[\min_{t \in [0,\ldots,T-1]} V^{\pi_t} - V^{\pi^e}\right] = \widetilde{O}\left(\frac{H^3\sqrt{dd_s + d^2}}{\sqrt{T}} + H\sqrt{\frac{\ln(TH|\mathcal{F}|)}{N}}\right).$$

We extend the above result to infinite dimensional RKHS below, where the dimension d in the above corollary is replaced by the intrinsic dimension, which can be bounded for RBF, Matern, and other common kernels (see Srinivas et al. (2009) for more details).

The following proposition shows that the bonus designed in Example 2 is valid.

Proposition 10 (KNR Bonus)**.** *Fix $\delta \in (0,1)$. With probability at least $1-\delta$, for all $t \in \mathbb{N}$, we have:*

$$\left\|\widehat{P}_t(\cdot|s,a) - P^\star(\cdot|s,a)\right\|_1 \leq \min\left\{\beta_t \|\phi(s,a)\|_{\Sigma_t^{-1}}, 2\right\}, \forall s,a,$$

where $\beta_t = \sqrt{2\lambda\|W^\star\|_2^2 + 8\sigma^2\left(d_s\ln(5) + 2\ln(t^2/\delta) + \ln(4) + \ln\left(\det(\Sigma_t)/\det(\lambda I)\right)\right)}$.

Proof. The proof directly follows the confidence ball construction and proof from Kakade et al. (2020a). Specifically, from Lemma B.5 in Kakade et al. (2020a), we have that with probability at least $1-\delta$, for all t:

$$\left\|\left(\widehat{W}_t - W^\star\right)(\Sigma_t)^{1/2}\right\|_2^2 \leq \beta_t.$$

Thus, with Lemma 16, we have:

$$\begin{aligned}\left\|\widehat{P}_t(\cdot|s,a) - P^\star(\cdot|s,a)\right\|_1 &\leq \frac{1}{\sigma}\left\|(\widehat{W}_t - W^\star)\phi(s,a)\right\|_2 \\ &\leq \left\|(\widehat{W}_t - W^\star)(\Sigma_t)^{1/2}\right\| \|\phi(s,a)\|_{\Sigma_t^{-1}}/\sigma \\ &\leq \frac{\beta_t}{\sigma}\|\phi(s,a)\|_{\Sigma_t^{-1}}.\end{aligned}$$

This concludes the proof. □

The following proposition bounds the information gain quantity.

Proposition 11 (Information Gain on KNRs)**.** *For simplicity, let us assume $\phi : \mathcal{S}\times\mathcal{A} \mapsto \mathbb{R}^d$, i.e., $\phi(s,a)$ is a d-dim feature vector. In this case, we will have:*

$$\mathcal{I}_T = O\left(H\left(d\ln(T^2/\delta) + dd_s + d^2\ln\left(1+\|W^\star\|_2^2 TH/\sigma^2\right)\right)\ln\left(1+\|W^\star\|_2^2TH/\sigma^2\right)\right).$$

Proof. From the previous proposition, we know that $\sigma_t^2(s,a) = (\beta_t^2/\sigma^2)\|\phi(s,a)\|^2_{\Sigma_t^{-1}}$. Setting $\lambda = \sigma^2/\|W^\star\|_2^2$, we will have $\beta_t^2/\sigma^2 \geq 1$, which means that $\min\{\sigma_t^2(s,a), 1\} \leq (\beta_t^2/\sigma^2)\min\left\{\|\phi(s,a)\|^2_{\Sigma_t^{-1}}, 1\right\}$.

Note that β_t is non-decreasing with respect to t, so $\beta_t \leq \beta_T$ for $T \geq t$, where

$$\beta_T = \sqrt{2\sigma^2 + 8\sigma^2(d_s\ln(5) + 2\ln(T^2/\delta) + \ln(4) + d\ln(1+TH\|W^\star\|_2^2/\sigma^2))}$$

Also we have $\sum_{t=0}^{T-1}\sum_{h=0}^{H-1}\min\left\{\|\phi(s_h^t,a_h^t)\|^2_{\Sigma_t^{-1}}, 1\right\} \leq H\sum_{t=0}^{T-1}\min\left\{\sum_{h=0}^{H-1}\|\phi(s_h^t,a_h^t)\|^2_{\Sigma_t^{-1}}, 1\right\}$, since $\min\{a_1,b_1\}+\min\{a_2,b_2\} \leq \min\{a_1+a_2, b_1+b_2\}$. Now call Lemma B.6 in Kakade et al. (2020a), we have:

$$\sum_{t=0}^{T-1}\min\left\{\sum_{h=0}^{H-1}\|\phi(s_h^t,a_h^t)\|^2_{\Sigma_t^{-1}}, 1\right\} \leq 2\ln\left(\det(\Sigma_T)/\det(\lambda I)\right) = 2d\ln\left(1+TH\|W^\star\|_2^2/\sigma^2\right). \tag{4}$$

Finally recall the definition of $\mathcal{I}_T$, we have:

$$\begin{aligned}\mathcal{I}_T &= \sum_{t=0}^{T-1}\sum_{h=0}^{H-1}\min\left\{\sigma_t^2(s_h^t,a_h^t),1\right\} \\ &\leq \frac{\beta_T^2}{\sigma^2}\sum_{t=0}^{T-1}\sum_{h=0}^{H-1}\min\left\{\|\phi(s_h^t,a_h^t)\|_{\Sigma_t^{-1}},1\right\} \\ &\leq \frac{\beta_T^2}{\sigma^2}2Hd\ln(1+\|W^\star\|_2^2TH/\sigma^2) \\ &\leq 2Hd\left(2+8\left(d_s\ln(5)+2\ln(T^2/\delta)+\ln(4)+d\ln\left(1+\|W^\star\|_2^2TH/\sigma^2\right)\right)\right)\ln\left(1+\|W^\star\|_2^2TH/\sigma^2\right) \\ &= H\left(4d+32dd_s+32d\ln(T^2/\delta)+32d+2d^2\ln\left(1+\|W^\star\|_2^2TH/\sigma^2\right)\right)\ln\left(1+\|W^\star\|_2^2TH/\sigma^2\right),\end{aligned}$$

which concludes the proof. □

Extension to Infinite Dimensional RKHS When $\phi : \mathcal{S}\times\mathcal{A}\mapsto\mathcal{H}$ where $\mathcal{H}$ is some infinite dimensional RKHS, we can bound our regret using the following intrinsic dimension:

$$\widetilde{d} = \max_{\{\{s_h^t,a_h^t\}_{h=0}^{H-1}\}_{t=0}^{T-1}}\ln\left(I + \frac{1}{\lambda}\sum_{t=0}^{T-1}\sum_{h=0}^{H-1}\phi(s_h^t,a_h^t)\phi(s_h^t,a_h^t)^\top\right).$$

In this case, recall Proposition 10, we have:

$$\beta_t \leq \beta_T \leq \sqrt{2\lambda\|W^\star\|_2^2 + 8\sigma^2\left(d_s\ln(5) + 2\ln(t^2/\delta) + \ln(4) + \ln\left(\det(\Sigma_T)/\det(\lambda I)\right)\right)}$$
$$\leq \sqrt{2\lambda\|W^\star\|_2^2 + 8\sigma^2\left(d_s\ln(5) + 2\ln(t^2/\delta) + \ln(4) + \widetilde{d}\right)}.$$

Also recall Eq. (4), we have:

$$\sum_{t=0}^{T-1}\min\left\{\sum_{h=0}^{H-1}\|\phi(s_h^t,a_h^t)\|^2_{\Sigma_t^{-1}}, 1\right\} \leq 2\ln\left(\det(\Sigma_T)/\det(\lambda I)\right) \leq 2\widetilde{d}.$$

Combine the above two, following similar derivation we had for finite dimensional setting, we have:

$$\mathcal{I}_T = \widetilde{O}\left(H\widetilde{d}^2 + H\widetilde{d}d_s\right).$$

A.2 General Function Class $\mathcal{G}$ with Bounded Eluder dimension

Proposition 12. *Fix $\delta \in (0,1)$. Consider a general function class G where G is discrete, and $\sup_{g\in\mathcal{G},s,a}\|g\|_2 \leq G$. At iteration t, denote $\widehat{g}_t \in \arg\min_{g\in\mathcal{G}}\sum_{i=0}^{t-1}\sum_{h=0}^{H-1}\|g(s_h^i,a_h^i) - s_{h+1}^i\|_2^2$, and denote a version space $\mathcal{G}_t$ as:*

$$\mathcal{G}_t = \left\{g\in\mathcal{G} : \sum_{i=0}^{t-1}\sum_{h=0}^{H-1}\left\|g(s_h^i,a_h^i) - \widehat{g}_t(s_h^i,a_h^i)\right\|_2^2 \leq c_t\right\},\ \text{with } c_t = \sigma G\sqrt{\ln(2t^2|\mathcal{G}|/\delta)tH}.$$

The with probability at least $1-\delta$, we have that for all t, and all s,a:

$$\left\|\widehat{P}_t(\cdot|s,a) - P^\star(\cdot|s,a)\right\|_1 \leq \min\left\{\frac{1}{\sigma}\max_{g_1\in\mathcal{G}_t,g_2\in\mathcal{G}_t}\|g_1(s,a) - g_2(s,a)\|_2, 2\right\}.$$

Proof. Consider a fixed function $g\in\mathcal{G}$. Let us denote $z_h^t = \left\|g(s_h^t,a_h^t) - s_{h+1}^t\right\|_2^2 - \left\|g^\star(s_h^t,a_h^t) - s_{h+1}^t\right\|_2^2$. We have:

$$z_h^t = \left(g(s_h^t,a_h^t) - g^\star(s_h^t,a_h^t)\right)^\top\left(g(s_h^t,a_h^t) + g^\star(s_h^t,a_h^t) - 2g^\star(s_h^t,a_h^t) - 2\epsilon_h^t\right)$$
$$= \left\|g(s_h^t,a_h^t) - g^\star(s_h^t,a_h^t)\right\|_2^2 - 2(g(s_h^t,a_h^t) - g^\star(s_h^t,a_h^t))^\top\epsilon_h^t.$$

Since $\epsilon_h^t \sim \mathcal{N}(0,\sigma^2 I)$, we must have:

$$2(g(s_h^t,a_h^t) - g^\star(s_h^t,a_h^t))^\top\epsilon_h^t \sim \mathcal{N}(0, 4\sigma^2\left\|g(s_h^t,a_h^t) - g^\star(s_h^t,a_h^t)\right\|_2^2)$$

Since $\sup_{g,s,a}\|g(s,a)\|_2 \leq G$, then $2(g(s_h^t,a_h^t) - g^\star(s_h^t,a_h^t))^\top\epsilon_h^t$ is a $2\sigma G$ sub-Gaussian random variable.

Call Lemma 3 in Russo & Van Roy (2014), we have that with probability at least $1-\delta$:

$$\sum_t\sum_h\left\|g(s_h^t,a_h^t) - s_{h+1}^t\right\|_2^2 \geq \sum_t\sum_h\left\|g^\star(s_h^t,a_h^t) - s_{h+1}^t\right\|_2^2$$
$$+ 2\sum_t\sum_h\left\|g(s_h^t,a_h^t) - g^\star(s_h^t,a_h^t)\right\|_2^2 - 4\sigma^2G^2\ln(1/\delta).$$

Note that the above can also be derived directly from Azuma-Bernstein's inequality. With a union bound over all $g\in\mathcal{G}$, we have:

$$\sum_t\sum_h\left\|g(s_h^t,a_h^t) - s_{h+1}^t\right\|_2^2 \geq \sum_t\sum_h\left\|g^\star(s_h^t,a_h^t) - s_{h+1}^t\right\|_2^2$$
$$+ 2\sum_t\sum_h\left\|g(s_h^t,a_h^t) - g^\star(s_h^t,a_h^t)\right\|_2^2 - 4\sigma^2G^2\ln(|\mathcal{G}|/\delta).$$

Set $g = \widehat{g}_t$, and use the fact that g_t is the minimizer of $\sum_t \sum_h \|g(s_h^t, a_h^t) - s_{h+1}^t\|_2^2$, we must have:

$$\sum_t \sum_h \left\| \widehat{g}_t(s_h^t, a_h^t) - g^\star(s_h^t, a_h^t) \right\|_2^2 \leq 2\sigma^2 G^2 \ln(2t^2|\mathcal{G}|/\delta).$$

Namely we prove that our version space $\mathcal{G}_t$ contains $g^\star$ for all t. Thus, we have:

$$\left\| \widehat{P}_t(\cdot|s,a) - P^\star(\cdot|s,a) \right\|_1 \leq \frac{1}{\sigma} \|\widehat{g}_t(s,a) - g^\star(s,a)\|_2 \leq \frac{1}{\sigma} \sup_{g_1 \in \mathcal{G}_t, g_2 \in \mathcal{G}_t} \|g_1(s,a) - g_2(s,a)\|_2,$$

where the last inequality holds since both $g^\star$ and $\widehat{g}_t$ belongs to the version $\mathcal{G}_t$.

□

Now we bound the information gain $\mathcal{I}_T$ below. The proof mainly follows from the proof in Osband & Van Roy (2014).

Lemma 13 (Lemma 1 in Osband & Van Roy (2014)). *Denote $\beta_t = 2\sigma^2 G^2 \ln(t^2|\mathcal{G}|/\delta)$. Let us denote the uncertainty measure $w_{t;h} = \sup_{f_1, f_2 \in \mathcal{G}_t} \|f_1(s_h^t, a_h^t) - f_2(s_h^t, a_h^t)\|_2$ (note that $w_{t;h}$ is non-negative). We have:*

$$\sum_{i=0}^{t-1} \sum_{h=0}^{H-1} \mathbf{1}\{w_{t;h}^2 > \epsilon\} \leq \left(\frac{4\beta_t}{\epsilon} + H \right) d_E(\sqrt{\epsilon}).$$

Proposition 14 (Bounding $\mathcal{I}_T$). *Denote $d = d_E(1/TH)$. We have*

$$\mathcal{I}_T = \left(1/\sigma^2 + HdG^2/\sigma^2 + 8G^2 \ln(T^2|\mathcal{G}|/\delta) d \ln(TH) \right).$$

Proof. Note that the uncertainty measures $w_{t;h}$ are non-negative. Let us reorder the sequence and denote the ordered one as $w_1 \geq w_2 \geq w_3 \cdots \geq w_{TH-H}$. For notational simplicity, denote $M = TH - H$ We have:

$$\sum_{i=0}^{T-1} \sum_{h=0}^{H-1} w_{t;h}^2 = \sum_{i=0}^{M-1} w_i^2 = 1 + \sum_i w_i^2 \mathbf{1}\{w_i^2 > \frac{1}{M}\}.$$

Consider any w_t where $w_t^2 \geq 1/M$. In this case, we know that $w_1^2 \geq w_2^2 \geq \cdots \geq w_t^2 \geq 1/M$. This means that:

$$t \leq \sum_i \sum_h \mathbf{1}\{w_{t;h}^2 > w_t^2\} \leq \left(\frac{4\beta_T}{w_t^2} + H \right) d_E(\sqrt{w_t}) \leq \left(\frac{4\beta_T}{w_t^2} + H \right) d_E(1/M),$$

where the second inequality uses the lemma above, and the last inequality uses the fact that $d_E(\epsilon)$ is non-decreasing when ϵ gets smaller. Denote $d = d_E(1/M)$. We have that $w_t^2 \leq \frac{4\beta_T d}{t - Hd}$. This means that for any $w_t^2 \geq 1/M$, we must have $w_t^2 \leq 4\beta_T d/(t - Hd)$. Thus, we have:

$$\begin{aligned}
\sum_{i=0}^{T-1} \sum_{h=0}^{H-1} w_{t;h}^2 &\leq 1 + HdG^2 + \sum_{\tau = Hd+1}^{M} w_\tau^2 \mathbf{1}\{w_\tau \geq 1/M\} \leq 1 + HdG^2 + 4\beta_T d \ln(M) \\
&\leq 1 + HdG^2 + 4\beta_T d \ln(TH)
\end{aligned}$$

Finally, recall the definition of $\mathcal{I}_T$, we have:

$$\begin{aligned}
\sum_{t=0}^{T-1} \sum_{h=0}^{H-1} \min\{\sigma_t^2(s_h^t, a_h^t), 1\} &\leq \sum_{t=0}^{T-1} \sigma_t^2(s_h^t, a_h^t) \\
&\leq \frac{1}{\sigma^2} \sum_{t=0}^{T-1} \sum_{h=0}^{H-1} w_{t;h}^2 \\
&\leq \frac{1}{\sigma^2} \left(1 + HdG^2 + 4\beta_T d \ln(TH) \right).
\end{aligned}$$

This concludes the proof. □

A.3 Proof of Theorem 5

This section provides the proof of Theorem 5.

Proof of Theorem 5. Below, we will construct A many MAB instances where each instance has A many arms and each arm has a Gaussian reward distribution with the fixed variance σ^2. Each of the A instance has the maximum mean reward equal to Δ, i.e., all these A instances have the same maximum arm mean reward. Consider any algorithm Alg that maps Δ together with the history of the interactions $\mathcal{H}_t = \{a_0, r_0, a_1, r_1, \ldots, a_{t-1}, r_{t-1}\}$ to a distribution over A actions. We will show for any such algorithm alg that knows Δ, with constant probability, there must exist a MAB instance from the K many MAB instances, such that Alg suffers at least $\Omega(\sqrt{AT})$ regret where T is the number of iterations.

Now we construct the A instances as follows. Consider the i-th instance ($i = 1, \ldots, A$). For arm j in the i-th instance, we define its mean as $\mu^i_j = \mathbf{1}\{i = j\}\Delta$. Namely, for MAB instance i, its arms have mean reward zero everywhere except that the i-th arm has reward mean Δ. Note that all these MAB instances have the same maximum mean reward, i.e., Δ. Hence, we cannot distinguish them by just revealing Δ to the learner.

We will construct an additional MAB instance (we name it as 0-th MAB instance) whose arms have reward mean zero. Note that this MAB instance has maximum mean reward 0 which is different from the previous A MAB instances that we constructed. However, we will only look at the regret of Alg on the previously constructed A MAB instances. I.e., we do not care about the regret of $\mathrm{Alg}(\Delta, \mathcal{H}_t)$ on the 0-th MAB instance.

Let us denote $\mathbb{P}_i$ (for $i = 0, \ldots, A$) as the distribution of the outcomes of algorithm $\mathrm{Alg}(\Delta, \mathcal{H}_t)$ interacting with MAB instance i for T iterations, and $\mathbb{E}_j[N_i(T)]$ as the expected number of times arm i is pulled by $\mathrm{Alg}(\Delta, \mathcal{H}_t)$ in MAB instance j. Consider MAB instance i with $i \geq 1$:

$$\mathbb{E}_i[N_i(T)] - \mathbb{E}_0[N_i(T)] \leq T \left\|\mathbb{P}_i - \mathbb{P}_0\right\|_1 \leq T\sqrt{\mathrm{KL}(\mathbb{P}_0, \mathbb{P}_i)} \leq T\sqrt{\Delta^2 \mathbb{E}_0[N_i(T)]},$$

where the last step uses the fact that we are running the same algorithm $\mathrm{Alg}(\Delta, \mathcal{H}_t)$ on both instance 0 and instance i (i.e., same policy for generating actions), and thus, $\mathrm{KL}(\mathbb{P}_0, \mathbb{P}_i) = \sum_{j=1}^{A} \mathbb{E}_0[N_j(T)] \mathrm{KL}\left(q_0(j), q_i(j)\right)$ (Lemma 15.1 in Lattimore & Szepesvári (2020)), where $q_i(j)$ is the reward distribution of arm j at instance i. Also recall that for instance 0 and instance i, their rewards only differ at arm i.

This implies that:

$$\mathbb{E}_i[N_i(T)] \leq \mathbb{E}_0[N_i(T)] + T\sqrt{\Delta^2 \mathbb{E}_0[N_i(T)]}.$$

Sum over $i = 1, \ldots, A$ on both sides, we have:

$$\begin{aligned}\sum_{i=1}^{A} \mathbb{E}_i[N_i(T)] &\leq T + T\sum_{i=1}^{A}\sqrt{\Delta^2 \mathbb{E}_0[N_i(T)]} \leq T + T\sqrt{A}\sqrt{\sum_{i=1}^{A}\Delta^2 \mathbb{E}_0[N_i(T)]} \\ &\leq T + T\sqrt{A}\sqrt{\Delta^2 T}\end{aligned}$$

Now let us calculate the regret of $\mathrm{Alg}(\Delta, \mathcal{H}_t)$ on i-th instance, we have:

$$R_i = T\Delta - \mathbb{E}_i[N_i(T)]\Delta.$$

Sum over $i = 1, \ldots, A$, we have:

$$\sum_{i=1}^{A} R_i = \Delta\left(AT - \sum_{i=1}^{A}\mathbb{E}_i[N_i(T)]\right) \geq \Delta\left(AT - T - T\sqrt{A\Delta^2 T}\right)$$

Set $\Delta = c\sqrt{A/T}$ for some c that we will specify later, we get:

$$\sum_{i=1}^{A} R_i \geq c\sqrt{\frac{A}{T}}\left(AT - T - cAT\right).$$

Set $c = 1/4$, we get:

$$\sum_{i=1}^{A} R_i \geq c\sqrt{\frac{A}{T}}\,(AT - T - cAT) \geq \frac{1}{4}\sqrt{AT}\,(A - 1 - A/4)$$
$$= \frac{1}{4}\sqrt{AT}\,(3A/4 - 1) \geq \frac{1}{4}\sqrt{AT}\,(A/4)\,,$$

assuming $A \geq 2$.

Thus there must exist $i \in \{1, \ldots, A\}$, such that:

$$R_i \geq \frac{1}{16}\sqrt{AT}.$$

Note that the above construction considered any algorithm $\mathrm{Alg}(\Delta, \mathcal{H}_t)$ that maps Δ and history to action distributions. Thus it concludes the proof. □

B Auxiliary Lemmas

Lemma 15 (Simulation Lemma)**.** *Consider any two functions $f : \mathcal{S} \times \mathcal{A} \mapsto [0, 1]$ and $\widehat{f} : \mathcal{S} \times \mathcal{A} \mapsto [0, 1]$, any two transitions P and $\widehat{P}$, and any policy $\pi : \mathcal{S} \mapsto \Delta(\mathcal{A})$. We have:*

$$V^{\pi}_{P;f} - V^{\pi}_{\widehat{P},\widehat{f}} = \sum_{h=0}^{H-1} \mathbb{E}_{s,a\sim d^{\pi}_{P}} \left[f(s,a) - \widehat{f}(s,a) + \mathbb{E}_{s'\sim P(\cdot|s,a)} V^{\pi}_{\widehat{P},\widehat{f};h}(s') - \mathbb{E}_{s'\sim \widehat{P}(\cdot|s,a)} V^{\pi}_{\widehat{P},\widehat{f};h}(s') \right]$$
$$\leq \sum_{h=0}^{H-1} \mathbb{E}_{s,a\sim d^{\pi}_{P}} \left[f(s,a) - \widehat{f}(s,a) + \|V^{\pi}_{\widehat{P},\widehat{f};h}\|_{\infty} \|P(\cdot|s,a) - \widehat{P}(\cdot|s,a)\|_1 \right].$$

where $V^{\pi}_{P,f;h}$ denotes the value function at time step h, under π, P, f.

Such simulation lemma is standard in model-based RL literature and the derviation can be found, for instance, in the proof of Lemma 10 from Sun et al. (2019a).

Lemma 16. *Consider two Gaussian distribution $P_1 := \mathcal{N}(\mu_1, \sigma^2 I)$ and $P_2 := \mathcal{N}(\mu_2, \sigma^2 I)$. We have:*

$$\|P_1 - P_2\|_1 \leq \frac{1}{\sigma} \|\mu_1 - \mu_2\|_2 \,.$$

C Implementation Details

C.1 Practical Instantiation of MobILE

We present a brief practical instantiation MobILE's components with details in Appendix Section C.

Dynamics model learning: We employ Gaussian Dynamics Models parameterized by an MLP (Rajeswaran et al., 2020), i.e., $\widehat{P}(s,a) := \mathcal{N}(h_\theta(s,a), \sigma^2 I)$, where, $h_\theta(s,a) = s + \sigma_{\Delta_s} \cdot \mathrm{MLP}_\theta(s_c, a_c)$, where, θ are MLP's trainable parameters, $s_c = (s - \mu_s)/\sigma_s$, $a_c = (a - \mu_a)/\sigma_a$. Next, for $(s, a, s') \in \mathcal{D}$, $\Delta_s = s' - s$ and σ_{Δ_s} is the standard deviation of the state differences $\Delta_s \in \mathcal{D}$. Finally, we use SGD with momentum (Sutskever et al., 2013) for purposes of training the parameters θ of the MLP.
Discriminator parameterization: We utilize MMD as our choice of IPM and define the discriminator as $f(s) = w^\top \psi(s)$, where, $\psi(s)$ are the Random Fourier Features (Rahimi & Recht, 2008).
Bonus parameterization: We utilize the discrepancy between predictions of an ensemble of a pair of dynamics models $h_{\theta_1}(s,a)$ and $h_{\theta_2}(s,a)$ for designing the bonus. Denote the disagreement at any (s,a) as $\delta(s,a) = \|h_{\theta_1}(s,a) - h_{\theta_2}(s,a)\|_2$. For a replay buffer $\mathcal{D}$, $\delta_{\mathcal{D}} = \max_{(s,a)\sim\mathcal{D}} \delta(s,a)$ is the maximum discrepancy. We set bonus as $b(s,a) = \min(\delta(s,a)/\delta_{\mathcal{D}}, 1) \cdot \lambda$ where $\lambda > 0$ is a tunable parameter. The normalization $1/\delta_{\mathcal{D}}$ helps the maintaining the bonus magnitude between $(0, 1]$.
PG oracle: We use TRPO (Schulman et al., 2015) to perform policy optimization inside the learned model.

Algorithm 2 `MobILE`: Model-based Imitation Learning and Exploring for ILFO (used in practical implementation)

```
1: **Require**: Expert Dataset $\mathcal{D}_e$, Access to dynamics of the true environment i.e. $P^\star$.
2: Initialize Policy $\pi_0$, Discriminator $w_0$, Replay Buffer of pre-determined size $\mathcal{D}$, Dynamics Model $\widehat{P}_{-1}$, Bonus $b_{-1}$.
3: **for** $t = 0, \cdots, T-1$ **do**
4:     **Online Interaction**: Execute $\pi_t$ in true environment $P^\star$ to get samples $\mathcal{S}_t$.
5:     **Update replay buffer**: $\mathcal{D} =$ Replay-Buffer-Update$(\mathcal{D}, \mathcal{S}_t)$ (refer to section C.3.2).
6:     **Update dynamics model**: Obtain $\widehat{P}_t$ by starting at $\widehat{P}_{t-1}$ and update using replay buffer $\mathcal{D}$ (refer to section C.3.1).
7:     **Bonus Update**: Update bonus $b_t : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^+$ using replay buffer $\mathcal{D}$ (refer to section C.3.3).
8:     **Discriminator Update**: Update discriminator as $w_t \leftarrow \max_w L(w; \pi_t, \widehat{P}_t, b_t, \mathcal{D}_e)$ (refer to section C.3.4).
9:     **Policy Update**: Perform incremental policy update through approx. minimization of $L(\cdot)$,
                       i.e.: $\pi_t \leftarrow \arg\min_\pi L(\pi; w_t, \widehat{P}_t, b_t, \mathcal{D}_e)$ by running $K_{PG}$ steps of TRPO
                       (refer to section C.3.5).
10: **end for**
11: **Return** $\pi_T$.
```

C.2 Environment Setup

This section sketches the details of how we setup the environments. We utilize the standard environment horizon of $500, 50, 200$ for `Cartpole-v1`, `Reacher-v2`, `Cartpole-v0`. For `Swimmer-v2` and `Hopper-v2`, we work with the environment horizon set to 400 (Kurutach et al., 2018; Nagabandi et al., 2018; Luo et al., 2018; Rajeswaran et al., 2020). Furthermore, for `Hopper-v2`, we add the velocity of the center of mass to the state parameterization (Rajeswaran et al., 2020; Luo et al., 2018). As noted in the main text, the expert policy is trained using NPG/TRPO (Kakade, 2001; Schulman et al., 2015) until it hits a value of (approximately) $460, -10, 38, 3000, 181$ for `Cartpole-v1`, `Reacher-v2`, `Swimmer-v2`, `Hopper-v2`, `Cartpole-v0` respectively. All the results presented in the experiments section are averaged over three seeds. Furthermore, in terms of baselines, we compare `MobILE` to BC. Note that BC has access to expert actions whereas our algorithm does not have access to the expert actions. For fair comparison, we use the same policy architecture for both `MobILE` and BC. We report the average of the best performance offered by BC for 500 epochs of training when run with 3 seeds, even if this occurs at different epochs for each of the runs. Furthermore, we used 2 CPUs with 16-32 GB of RAM usage to perform all our benchmarking runs implemented in Pytorch. Finally, we build our codebase on top of Open-AI's implementation of TRPO (Dhariwal et al., 2017) for environments with discrete actions, and the MJRL repository (Rajeswaran et al., 2017) for continuous action environments.

C.3 Practical Implementation of `MobILE`

We will begin with presenting the implementation details of `MobILE` (refer to Algorithm 2):

C.3.1 Dynamics Model Training

As detailed in the main paper, we utilize a class of Gaussian Dynamics Models parameterized by an MLP (Rajeswaran et al., 2020), i.e. $s' \sim s + \mathcal{N}(h_\theta(s_c, a_c), \sigma^2 \cdot I)$, where, $h_\theta(\cdot)$ is an MLP with trainable parameters θ; s_c, a_c are centered and normalized state and actions respectively; note that we predict normalized state differences instead of the next state directly.

In practice, we fine tune our estimate of dynamics models based on the new contents of the replay buffer as opposed to re-training the models from scratch, which is computationally more expensive. In particular, we start from the estimate $\widehat{P}_{t-1}$ in the $t-1$ epoch and perform multiple updates gradient updates using the contents of the replay buffer $\mathcal{D}$. We utilize constant stepsize SGD with momentum (Sutskever et al., 2013) for updating our dynamics models. Furthermore, since the distribution of (s, a, s') pairs continually drift as the algorithm progresses (for instance, because we observe a new state), we utilize gradient clipping to ensure our model does not diverge due to the aggressive nature of our updates.

C.3.2 REPLAY BUFFER

Since we perform incremental training of our dynamics model, we utilize a replay buffer of a fixed size rather than training our dynamics model on all previously collected online (s, a, s') samples. Note that the replay buffer could contain data from all prior online interactions should we re-train our dynamics model from scratch at every epoch.

C.3.3 DESIGN OF BONUS FUNCTION

We utilize an ensemble of two transition dynamics models incrementally learned using the contents of the replay buffer. Specifically, given the models $h_{\theta_1}(\cdot)$ and $h_{\theta_2}(\cdot)$, we compute the discrepancy as: $\delta(s,a) = ||h_{\theta_1}(s,a) - h_{\theta_2}(s,a)||_2$. Moreover, given a replay buffer $\mathcal{D}$, we compute the maximum discrepancy as $\delta_{\mathcal{D}} = \max_{(s,a,s')\sim\mathcal{D}} \delta(s,a)$. We then set the bonus as $b(s,a) = \min\left(1, \delta(s,a)/\delta_{\mathcal{D}}\right) \cdot \lambda$, thus ensuring the magnitude of our bonus remains bounded between $[0, \lambda]$.

C.3.4 DISCRIMINATOR UPDATE

Recall that $f_w(s) = w^\top \psi(s)$, where w are the parameters of the discriminator. Given a policy π, the update for the parameters w take the following form:

$$\max_{w:||w||_2^2 \le \zeta} L(w; \pi, \widehat{P}, b, \mathcal{D}_e) := \mathbb{E}_{(s,a)\sim d^\pi_{\widehat{P}}}\left[f_w(s) - b(s,a)\right] - \mathbb{E}_{s\sim\mathcal{D}_e}\left[f_w(s)\right]$$
$$\equiv \max_w L_\zeta(w; \pi, \widehat{P}, b, \mathcal{D}_e) = \mathbb{E}_{(s,a)\sim d^\pi_{\widehat{P}}}\left[f_w(s) - b(s,a)\right] - \mathbb{E}_{s\sim\mathcal{D}_e}\left[f_w(s)\right] - \frac{1}{2}\cdot\left(||w||_2^2 - \zeta\right),$$
$$\implies \partial_w L_\zeta(w; \pi, \widehat{P}, b, \mathcal{D}_e) = \mathbb{E}_{s\sim d^\pi_{\widehat{P}}}\left[\psi(s)\right] - \mathbb{E}_{s\sim\mathcal{D}_e}\left[\psi(s)\right] - w \in 0,$$

where, $\partial_w L_\zeta(w; \pi, \widehat{P}, b, \mathcal{D}_e)$ denotes the sub-differential of $L_\zeta(\cdot)$ wrt w. This implies:

1. **Exact Update:** $w^* = \mathcal{P}_{\mathcal{B}(\zeta)}\left(\mathbb{E}_{s\sim d^\pi_{\widehat{P}}}\left[\psi(s)\right] - \mathbb{E}_{s\sim\mathcal{D}_e}\left[\psi(s)\right]\right)$, $\mathcal{P}_.$ is the projection operator, and $\mathcal{B}(\zeta)$ is the $\zeta-$norm ball.
2. **Gradient Ascent Update:**
   $w_{t+1} = \mathcal{P}_{\mathcal{B}(\zeta)}\left((1-\eta_w)w_t + \eta_w \cdot \left(\mathbb{E}_{s\sim d^\pi_{\widehat{P}}}\left[\psi(s)\right] - \mathbb{E}_{s\sim\mathcal{D}_e}\left[\psi(s)\right]\right)\right)$, $\eta_w > 0$ is the step-size.

Empirically, we find either of the updates to work reasonably well. Furthermore, in certain cases, we empirically observe the gradient ascent update to yield more stability compared to the exact updates.

C.3.5 MODEL-BASED POLICY UPDATE

Once the maximization of the discriminator parameters w is performed, consider the policy optimization problem, i.e.,

$$\min_\pi L(\pi; w, \widehat{P}, b, \mathcal{D}_e) := \mathbb{E}_{(s,a)\sim d^\pi_{\widehat{P}}}\left[f_w(s) - b(s,a)\right] - \mathbb{E}_{s\sim\mathcal{D}_e}\left[f_w(s)\right]$$
$$\equiv \min_\pi L(\pi; w, \widehat{P}, b, \mathcal{D}_e) = \mathbb{E}_{(s,a)\sim d^\pi_{\widehat{P}}}\left[f_w(s) - b(s,a)\right]$$

Hence we perform model-based policy optimization under $\widehat{P}$ and cost function $f_w(s) - b(s,a)$. In practice, we perform approximate minimization of $L(\cdot)$ by incrementally updating the policy using K_{PG}-steps of policy gradient, where, K_{PG} is a tunable hyper-parameter. In our experiments, we find that setting K_{PG} to be around 10 to generally be a reasonable choice (for precise values, refer to Table 1). This paper utilizes TRPO (Schulman et al., 2015) as our choice of policy gradient method; note that this can be replaced by other alternatives including PPO (Schulman et al., 2017), SAC (Haarnoja et al., 2018) *etc.* Similar to practical implementations of existing policy gradient methods, we implement a reward filter by clipping the IPM reward $f(s)$ by truncating it between $c_{\min}$ and $c_{\max}$ as this leads to stability of the policy gradient updates. Note that the minimization is done with access to $\widehat{P}$,

Parameter	`Cartpole-v1`	`Reacher-v2`	`Cartpole-v0`	`Hopper-v2`
Environment Specifications				
Horizon H	500	50	200	400
Expert Performance ($\approx$)	460	-10	181	3000
# online samples per outer loop	$2 \cdot H$	$2 \cdot H$	$2 \cdot H$	$8 \cdot H$
Dynamics Model				
Architecture/Non-linearity	MLP$(64, 64)$/ReLU	MLP$(64, 64)$/ReLU	MLP$(64, 64)$/ReLU	MLP$(512, 512)$/ReLU
Optimizer(LR, Momentum, Batch Size)	SGD$(0.005, 0.99, 256)$	SGD$(0.005, 0.99, 256)$	SGD$(0.005, 0.99, 256)$	SGD$(0.005, 0.99, 256)$
# train passes per outer loop	20	100	20	50
Grad Clipping	2.0	2.0	2.0	4.0
Replay Buffer Size	$10 \cdot H$	$10 \cdot H$	$10 \cdot H$	$16 \cdot H$
Ensemble based bonus				
# models/bonus range	2/$[0, 1]$	2/$[0, 1]$	2/$[0, 1]$	2/$[0, 1]$
bonus weight λ (grid search values)– includes values for ablations	$\{0.00001, 0.0001, 0.001, 0.01\}$	$\{0.0005, 0.0015, 0.005, 0.015, 0.05, 0.15\}$	$\{0.0001, 0.001, 0.01\}$	$\{0.0001, 0.001, 0.01, 0.1, 0.5, 0.7\}$
IPM parameters				
Step size for w update (η_w)	Exact	Exact	Exact	Exact
# RFFs/BW Heuristic	128/0.1 quantile	128 / 0.1 quantile	128 / 0.1 quantile	128 / 0.1 quantile
Policy parameterization				
Architecture/Non-linearity	MLP$(64, 64)$/TanH	MLP$(64, 64)$/TanH	MLP$(32, 32)$/TanH	MLP$(32, 32)$/TanH
Policy Constraints	None	None	None	$\log \sigma_{\min} = -1.0$
Planning Algorithm				
# model samples per TRPO step	$2 \cdot H$	$10 \cdot H$	$4 \cdot H$	$8 \cdot H$
# TRPO steps per outer loop (K_{PG})	3	10	5	10
Reward Filter ($c_{\min}, c_{\max}$)	$[-1, 0]$	$[-1, 0]$	$[-1, 0]$	$[-1, 0]$
TRPO Parameters (CG iters, dampening, kl, gae$_\lambda$, γ)	$(50, 0.001, 0.01, 0.97, 0.995)$	$(100, 0.001, 0.01, 0.97, 0.995)$	$(100, 0.001, 0.01, 0.97, 0.995)$	$(10, 0.0001, 0.025, 0.97, 0.995)$
Critic parameterization				
Architecture/Non-linearity	MLP$(128, 128)$/ReLU	MLP$(128, 128)$/ReLU	MLP$(32, 32)$/ReLU	MLP$(128, 128)$/ReLU
Optimizer (LR, Batch Size, ϵ, Regularization)	Adam$(0.001, 64, 1e-5, 0)$	Adam$(0.001, 64, 1e-5, 0)$	Adam$(0.001, 64, 1e-5, 0)$	Adam$(0.001, 64, 1e-8, 1e-3)$
# train passes per TRPO update	1	1	1	2

Table 1: List of various Hyperparameters employed in `MobILE`'s implementation.

which implies we perform *model-based* planning. Empirically, for purposes of tuning the exploration-imitation parameter λ, we minimize a surrogate namely: $\mathbb{E}_{(s,a)\sim d^{\pi}_{\widehat{P}}}\left[(1-\lambda)\cdot f_w(s) - b(s,a)\right]$ (recall that $b(s,a)$ has a factor of λ associated with it). This ensures that we can precisely control the magnitude of the bonuses against the IPM costs, which, in our experience is empirically easier to work with.

C.4 Hyper-parameter Details

This section presents an overview of the list of hyper-parameters necessary to implement Algorithm 1 in practice, as described in Algorithm 2. The list of hyper-parameters is precisely listed out in table 1. The hyper-parameters are broadly categorized into ones corresponding to various components of `MobILE`, namely, (a) environment specifications, (b) dynamics model, (c) ensemble based bonus, (d) IPM parameterization, (e) Policy parameterization, (f) Planning algorithm parameters, (g) Critic parameterization. Note that if there a hyper-parameter that has not been listed, for instance, say, the value of momentum for the ADAM optimizer in the critic, this has been left as is the default value defined in Pytorch.