Provable Representation Learning for Imitation Learning via Bi-level Optimization

Sanjeev Arora [1 2] **Simon S. Du** [2] **Sham Kakade** [3] **Yuping Luo** [1] **Nikunj Saunshi** [1]

Abstract

A common strategy in modern learning systems is to learn a representation that is useful for many tasks, a.k.a. representation learning. We study this strategy in the imitation learning setting for Markov decision processes (MDPs) where multiple experts' trajectories are available. We formulate representation learning as a bi-level optimization problem where the "outer" optimization tries to learn the joint representation and the "inner" optimization encodes the imitation learning setup and tries to learn task-specific parameters. We instantiate this framework for the imitation learning settings of behavior cloning and observation-alone. Theoretically, we show using our framework that representation learning can provide sample complexity benefits for imitation learning in both settings. We also provide proof-of-concept experiments to verify our theory.

1. Introduction

Humans can often learn from experts quickly and with a few demonstrations and we would like our artificial agents to do the same. However, even for simple imitation learning tasks, the current state-of-the-art methods require thousand of demonstrations. Humans do not learn new skills from scratch. We can summarize learned skills, distill them and build a common ground, a.k.a, representation that is useful for learning future skills. Can we build an agent to do the same?

The current paper studies how to apply representation learning to imitation learning. Specifically, we want our agent to be able to learn a representation from multiple experts' demonstrations, where the experts aim to solve different Markov decision processes (MDPs) that share the same state and action spaces but can differ in the transition and reward functions. The agent can use this representation to reduce the number of demonstrations required for a new imitation learning task. While several methods have been proposed (Duan et al., 2017; Finn et al., 2017b; James et al., 2018) to build agents that can adapt quickly to new tasks, none of them, to our knowledge, give provable guarantees showing the benefit of using past experience. Furthermore, they do not focus on learning a representation. See Section 2 for more discussions.

In this work, we propose a framework to formulate this problem and analyze the statistical gains of representation learning for imitation learning. The main idea is to use bi-level optimization formulation where the "outer" optimization tries to learn the joint representation and the "inner" optimization encodes the imitation learning setup and tries to learn task-specific parameters. In particular, the inner optimization is flexible enough to allow the agent to interact with the environment. This framework allows us to do a rigorous analysis to show provable benefits of representation learning for imitation learning. With this framework at hand, we make the following concrete contributions:

- We first instantiate our framework in the setting where the agent can observe experts' actions and tries to find a policy that matches the expert's policy, a.k.a, behavior cloning. This setting can be viewed as a straightforward extension of multi-task representation learning for supervised learning (Maurer et al., 2016). We show in this setting that with sufficient number of experts (possibly optimizing for different reward functions), the agent can learn a representation that provably reduces the sample complexity for a new target imitation learning task.
- Next, we consider a more challenging setting where the agent *cannot* observe experts' actions but only their states, a.k.a., the observation-alone setting. We set the inner optimization as a min-max problem inspired by (Sun et al., 2019). Notably, this min-max problem requires the agent to interact with the environment to collect samples. We again show that with sufficient number of experts, the agent can learn a representation that provably reduces the sample complexity for a target task where the agent

[1]Princeton University, Princeton, New Jersey, USA [2]Institute for Advanced Study, Princeton, New Jersey, USA [3]University of Washington, Seattle, Washington, USA. Correspondence to: Nikunj Saunshi <nsaunshi@cs.princeton.edu>.

Proceedings of the 37^{th} International Conference on Machine Learning, Vienna, Austria, PMLR 119, 2020.

cannot observe actions from source and target experts.
- We conduct experiments in both settings to verify our theoretical insights by learning a representation from multiple tasks using our framework and testing it on a new task from the same setting. Additionally, we use these learned representations to learn a policy in the RL setting by doing policy optimization. We observe that by learning representations the agent can learn a good policy with fewer samples than needed to learn a policy from scratch.

The key contribution of our work is to connect existing literature on multi-task representation learning that deals with supervised learning (Maurer et al., 2016) to single task imitation learning methods with guarantees (Syed & Schapire, 2010; Ross et al., 2011; Sun et al., 2019). To our knowledge, this is the first work showing such guarantees for general losses that are not necessarily convex.

Organization: In Section 2, we review and discuss related work. Section 3 reviews necessary concepts and describes the basic representation learning setup. In Section 4, we formulate representation learning for imitation learning as a bi-level optimization problem and give an overview of the kind of results we prove. In Section 5, we show our theoretical guarantees for behavior cloning, i.e., the case when the agent can observe experts' actions. In Section 6, we discuss our theoretical result for the observation alone setting. In Section 7, we present our experimental results showing the benefit of representation learning for imitation learning via our framework. We conclude in Section 8 and defer technical proofs to appendix.

2. Related Work

Representation learning has shown its great power in various domains; see Bengio et al. (2013) for a survey. Theoretically, Maurer et al. (2016) studied the benefit representation learning for sample complexity reduction in the multi-task supervised learning setting. Recently, Arora et al. (2019) analyzed the benefit of representation learning via contrastive learning. While these papers all build representations for the agent / learner, researchers also try to build representations about the environment / physical world (Wu et al., 2017).

Imitation learning can help with sample efficiency of many problems (Ross & Bagnell, 2010; Sun et al., 2017; Daumé et al., 2009; Chang et al., 2015; Pan et al., 2018). Most existing work consider the setting where the learner can observe expert's action. A general strategy is use supervised learning to learn a policy that maps the state to action that matches expert's behaviors. The most straightforward one is behavior cloning (Pomerleau, 1991), which we also study in our paper. More advanced approaches have also been proposed (Ross et al., 2011; Ross & Bagnell, 2014; Sun et al., 2018). These approaches, including behavior cloning, often enjoy sound theoretical guarantees in the single task case. Our work extends the theoretical guarantees of behavior cloning to the multi-task representation learning setting.

This paper also considers a more challenging setting, imitation learning from observation alone. Though some model-based methods have been proposed (Torabi et al., 2018; Edwards et al., 2018), these methods lack theoretical guarantees. Another line of work learns a policy that minimizes the difference between the state distributions induced by it and the expert policy, under certain distributional metric (Ho & Ermon, 2016). Sun et al. (2019) gave a theoretical analysis to characterize the sample complexity of this approach and our method for this setting is inspired by their approach.

A line of work uses meta-learning for imitation learning (Duan et al., 2017; Finn et al., 2017b; James et al., 2018). Our work is different from theirs as we want to explicitly learn a representation that is useful across all tasks whereas these work try to learn a meta-algorithm that can quickly adapt to a new task. For example, Finn et al. (2017b) used a gradient based method for adaptation. Recently Raghu et al. (2019) argued that most of the power of MAML (Finn et al., 2017a) like approaches comes from learning a shared representation.

On the theoretical side of meta-learning and multi-task learning, Baxter (2000) performed the first theoretical analysis and gave sample complexity bounds using covering numbers. Maurer (2009) analyzed linear representation learning, while Bullins et al. (2019); Denevi et al. (2018) provide efficient algorithms to learn linear representations that can reduce sample complexity of a new task. Another recent line of work analyzes gradient based meta-learning methods, similar to MAML (Finn et al., 2017a). Existing work on the sample complexity and regret of these methods (Denevi et al., 2019; Finn et al., 2019; Khodak et al., 2019) show guarantees for convex losses by leveraging tools from online convex optimization. In contrast, our analysis works for arbitrary function classes and the bounds depend on the Gaussian averages of these classes. Recent work (Rajeswaran et al., 2019) uses a bi-level optimization framework for meta-learning and improves computation aspects of meta-learning through implicit differentiation; our interest lies in the statistical aspects.

3. Preliminaries

Markov Decision Processes (MDPs): Let $\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, C, \nu)$ be an MDP, where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the finite action space with $|\mathcal{A}| = K$, $H \in \mathbb{Z}_+$ is the planning horizon, $P : \mathcal{S} \times \mathcal{A} \to \triangle(\mathcal{S})$ is the transition function, $C : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is the cost function and $\nu \in \triangle(S)$ is the initial state distribution. We assume that cost is bounded by 1, i.e. $C(s, a) \leq 1, \forall s \in \mathcal{S}, a \in \mathcal{A}$. This is a stan-

dard regularity condition used in many theoretical reinforcement learning work. A (stochastic) policy is defined as $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_H)$, where $\pi_h : \mathcal{S} \to \triangle(\mathcal{A})$ prescribes a distribution over action for each state at level $h \in [H]$. For a stationary policy, we have $\pi_1 = \cdots = \pi_H = \pi$. A policy $\boldsymbol{\pi}$ induces a random trajectory $s_1, a_1, s_2, a_2, \ldots, s_H, a_H$ where $s_1 \sim \nu, a_1 \sim \pi_1(s), s_2 \sim P_{s_1,a_1}$ etc. Let $\nu_h^{\boldsymbol{\pi}}$ denote the distribution over $\mathcal{S}$ induced at level h by policy $\boldsymbol{\pi}$. The value function $V_h^{\boldsymbol{\pi}} : \mathcal{S} \to \mathbb{R}$ is defined as

$$V_h^{\boldsymbol{\pi}}(s_h) = \mathbb{E}\left[\sum_{i=h}^{H} C(s_i, a_i) \mid a_i \sim \pi_i(s_i), s_{i+1} \sim P_{s_i,a_i}\right]$$

and the state-action function $Q_h^{\boldsymbol{\pi}}(s_h, a_h)$ is defined as $Q_h^{\boldsymbol{\pi}}(s_h, a_h) = \mathbb{E}_{s_{h+1} \sim P_{s_h,a_h}}[V_h^{\boldsymbol{\pi}}(s_{h+1})]$. The goal is to learn a policy $\boldsymbol{\pi}$ that minimizes the *expected cost* $J(\boldsymbol{\pi}) = \mathbb{E}_{s_1 \sim \nu} V_1^{\boldsymbol{\pi}}(s_1)$. We define the Bellman operator at level h for any policy $\boldsymbol{\pi}$ as $\Gamma_h^{\boldsymbol{\pi}} : \mathbb{R}^{\mathcal{S}} \to \mathbb{R}^{\mathcal{S}}$, where for $s \in \mathcal{S}$ and $g \in \mathbb{R}^{\mathcal{S}}$,

$$(\Gamma_h^{\boldsymbol{\pi}} g)(s) \coloneqq \mathbb{E}_{a \sim \pi_h(s), s' \sim P_{s,a}}[g(s')] \tag{1}$$

Multi-task Imitation learning: We formally describe the problem we want to study. We assume there are multiple tasks (MDPs) sampled i.i.d. from a distribution η. A task $\mu \sim \eta$ is an MDP $\mathcal{M}_\mu = (\mathcal{S}, \mathcal{A}, H, P_\mu, C_\mu, \nu_\mu)$; all tasks share everything except the cost function, initial state distribution and transition function. For simplicity of presentation, we will assume a common transition function P for all tasks; proofs remain exactly the same even otherwise. For every task μ, $\boldsymbol{\pi}_\mu^* = (\pi_{1,\mu}^*, \ldots, \pi_{H,\mu}^*)$ is an *expert policy* that the learner has access to in the form of trajectories induced by that policy. The trajectories may or may not contain expert's actions. These correspond to two settings that we discuss in more detail in Section 5 and Section 6. The distributions of states induced by this policy at different levels are denoted by $\{\nu_{1,\mu}^*, \ldots, \nu_{H,\mu}^*\}$ and the average state distribution as $\nu_\mu^* = \frac{1}{H}\sum_{h=1}^{H} \nu_{h,\mu}^*$. We define $V_{h,\mu}^*$ to be the value function of $\boldsymbol{\pi}_\mu^*$ and J_μ to be the expected cost function for task μ. We will drop the subscript μ whenever the task at hand is clear from context. Of interest is also the special case where the expert policy $\boldsymbol{\pi}_\mu^*$ is stationary.

Representation learning: In this work, we wish to learn policies from a function class of the form $\Pi = \mathcal{F} \circ \Phi$, where $\Phi \subseteq \{\phi : \mathcal{S} \to \mathbb{R}^d \mid \|\phi(s)\|_2 \leq R\}$ is a class of bounded norm *representation functions* mapping states to vectors and $\mathcal{F} \subseteq \{f : \mathbb{R}^d \to \Delta(\mathcal{A})\}$ is a class of functions mapping state representations to distribution over actions. We will be using linear functions, i.e. $\mathcal{F} = \{x \to \texttt{softmax}(Wx) \mid W \in \mathbb{R}^{K \times d}, \|W\|_F \leq 1\}$. We denote a policy parametrized by $\phi \in \Phi$ and $f \in \mathcal{F}$ by $\pi^{\phi,f}$, where $\pi^{\phi,f}(a|s) = f(\phi(s))_a$. In some cases, we may also use the policy $\pi^{\phi,f}(a|s) = \mathbb{I}\{a = \arg\max_{a' \in A} f(\phi(s))_{a'}\}$[1]. Denote $\Pi^\phi = \{\pi^{\phi,f} : f \in \mathcal{F}\}$ to be the class of policies that use ϕ as the representation function.

Given demonstrations from expert policies for T tasks sampled independently from η, we wish to first learn representation functions $(\hat{\phi}_1, \ldots, \hat{\phi}_H)$ so that we can use a few demonstrations from an expert policy $\boldsymbol{\pi}^*$ for new task $\mu \sim \eta$ and learn a policy $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_H)$ that uses the learned representations, i.e. $\pi_h \in \Pi^{\hat{\phi}_h}$, such that has average cost of $\boldsymbol{\pi}$ is not too far away from $\boldsymbol{\pi}^*$. In the case of stationary policies, we need to learn a single ϕ by using tasks and learn $\pi \in \Pi^\phi$ for a new task. The hope is that data from multiple tasks can be used to learn a complicated function $\phi \in \Phi$ first, thus requiring only a few samples for a new task to learn a linear policy from the class Π^ϕ.

Gaussian complexity: As in (Maurer et al., 2016), we measure the complexity of a function class $\mathcal{H} \subseteq \{h : \mathcal{X} \to \mathbb{R}^d\}$ on a set $\mathbf{X} = (X_1, \ldots, X_n) \in \mathcal{X}^n$ by using the following Gaussian average

$$G(\mathcal{H}(\mathbf{X})) = \mathbb{E}\left[\sup_{h \in \mathcal{H}} \sum_{\substack{i=1 \\ j=1}}^{d,n} \gamma_{ij} h_i(X_j) \mid X_j\right] \tag{2}$$

where γ_{ij} are independent standard normal variables. Bartlett & Mendelson (2003) also used Gaussian averages to show some generalization bounds.

4. Bi-level Optimization Framework

In this section we introduce our framework and give a high-level description of the conditions under which this framework gives us statistical guarantees. Our main idea is to phrase learning representations for imitation learning as the following bi-level optimization

$$\min_{\phi \in \Phi} L(\phi) \coloneqq \mathop{\mathbb{E}}_{\mu \sim \eta} \min_{\pi \in \Pi^\phi} \ell^\mu(\pi) \tag{3}$$

Here ℓ^μ is the *inner* loss function that penalizes π being different from π_μ^* for the task μ. In general, one can use any loss ℓ^μ that is used for single task imitation learning, e.g. for the behavioral cloning setting (cf. Section 5), ℓ^μ is a classification like loss that penalizes the mismatch between predictions by π^* and π, while for the observation-alone setting (cf. Section 6) it is some measure of distance between the state visitation distributions induced by π and π^*. The *outer* loss function is over the representation ϕ. The use of bi-level optimization framework naturally enforces policies in the inner optimization to share the same representation.

While Equation 3 is formulated in terms of the distribution η, in practice we only have access to few samples for T tasks;

[1] Break ties in any way

let $\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(T)}$ denote samples from tasks $\mu^{(1)}, \ldots, \mu^{(T)}$ sampled i.i.d. from η. We thus learn the representation $\hat{\phi}$ by minimizing empirical version $\hat{L}$ of Equation 3.

$$\hat{L}(\phi) = \frac{1}{T}\sum_{i=1}^{T} \min_{\pi \in \Pi^{\phi}} \ell^{\mathbf{x}^{(i)}}(\pi) = \frac{1}{T}\sum_{i=1}^{T} \ell^{\mathbf{x}^{(i)}}(\pi^{\phi, \mathbf{x}^{(i)}})$$

where $\ell^{\mathbf{x}}$ is the empirical loss on samples $\mathbf{x}$ and $\pi^{\phi,\mathbf{x}} = \arg\min_{\pi \in \Pi^{\phi}} \ell^{\mathbf{x}}(\pi)$ corresponds to a task specific policy that uses a fixed representation ϕ. Our goal then is to show that for a new task $\mu \sim \eta$, the learned representation can be used to learn a policy $\pi^{\hat{\phi},\mathbf{x}}$ by using samples $\mathbf{x}$ from the task μ that has low expected MDP cost J_μ (defined in Section 3)

Informal Theorem 4.1. *With high probability over the sampling of train task data and with sufficient number of tasks and samples (expert demonstrations) per task, $\hat{\phi} = \arg\min_{\phi \in \Phi} \hat{L}(\phi)$ will satisfy*

$$\mathop{\mathbb{E}}_{\mu \sim \eta} \mathop{\mathbb{E}}_{\mathbf{x}} J_\mu(\pi^{\hat{\phi},\mathbf{x}}) - \mathop{\mathbb{E}}_{\mu \sim \eta} J_\mu(\pi^*_\mu) \text{ is small}$$

At a high level, in order to prove such a theorem for a particular choice of ℓ^μ, we would need to prove the following three properties about ℓ^μ and $\ell^{\mathbf{x}}$:

1. $\ell^{\mathbf{x}}(\pi)$ concentrates to $\ell^\mu(\pi)$ simultaneously for all $\pi \in \Pi^{\phi}$ (for a fixed ϕ), with sample complexity depending on some complexity measure of Π^{ϕ} rather than being polynomial in $|\mathcal{S}|$;
2. if ϕ and ϕ' induce "similar" representations then $\min_{\pi \in \Pi^{\phi}} \ell^\mu(\pi)$ and $\min_{\pi \in \Pi^{\phi'}} \ell^\mu(\pi)$ are close;
3. a small value of $\ell^\mu(\pi)$ implies a small value for $J_\mu(\pi) - J_\mu(\pi^*_\mu)$.

The first property ensures that learning a policy for a single task by fixing the representation is sample efficient, thus making representation learning a useful problem to solve. The second property is specific to representation learning and requires ℓ^μ to use representations in a smooth way. This ensures that the empirical loss for T tasks is a good estimate for the average loss on tasks sampled from η. The third property ensures that matching the behavior of the expert as measured by the loss ℓ^μ ensures low average cost i.e., ℓ^μ is meaningful for the average cost; any standard imitation learning loss will satisfy this. We *prove* these three properties for the cases where ℓ^μ is the either behavioral cloning loss or observation-alone loss, with natural choices for the empirical loss $\ell^{\mathbf{x}}$. However the general proof recipe can be used for potentially many other settings and loss functions.

In the next section, we will describe representation learning for behavioral cloning as an instantiation of the above framework and describe the various components of the framework. Furthermore we will describe the results and give a proof sketch to show how the aforementioned properties help us show our final guarantees. The guarantees for this setting follow almost directly from results in (Maurer et al., 2016) and (Ross et al., 2011). Later in Section 6 we describe the same for the observations alone setting which is more non-trivial.

5. Representation Learning for Behavioral Cloning

Choice of ℓ^μ: We first specify the inner loss function in the bi-level optimization framework. In the single task setting, the goal of behavioral cloning (Syed & Schapire, 2010; Ross et al., 2011) is to use expert trajectories of the form $\tau = (s_1, a_1, \ldots, s_H, a_H)$ to learn a stationary policy[2] that tries to mimic the decisions of the expert policy on the states visited by the expert. For a task μ, this reduces to a supervised classification problem that minimizes a surrogate to the following loss $\ell^\mu_{0-1}(\pi) = \mathbb{E}_{s \sim \nu^*_\mu, a \sim \pi^*_\mu(s)} \mathbb{I}\{\pi(s) \neq a\}$. We abuse notation and denote this distribution over (s, a) for task μ as μ; so $(s, a) \sim \mu$ is the same as $s \sim \nu^*_\mu, a \sim \pi^*_\mu(s)$. Prior work (Syed & Schapire, 2010; Ross et al., 2011) have shown that a small value of $\ell^\mu_{0-1}(\pi)$ implies a small difference $J(\pi) - J(\pi^*)$. Thus for our setting, we choose ℓ^μ to be of the following form

$$\ell^\mu(\pi) = \mathop{\mathbb{E}}_{s \sim \nu^*_\mu, a \sim \pi^*_\mu(s)} \ell(\pi(s), a) = \mathop{\mathbb{E}}_{(s,a) \sim \mu} \ell(\pi(s), a) \quad (4)$$

where ℓ is any surrogate to 0-1 loss $\mathbb{I}\{a \neq \arg\max_{a' \in A} \pi(s)_{a'}\}$ that is *Lipschitz* in $\phi(s)$. In this work we consider the logistic loss $\ell(\pi(s), a) = -\log(\pi(s)_a)$.

Learning ϕ from samples: Given expert trajectories for T tasks $\mu^{(1)}, \ldots, \mu^{(T)}$ we construct a dataset $\mathbf{X} = \{\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(T)}\}$, where $\mathbf{x}^{(t)} = \{(s^t_j, a^t_j)\}_{j=1}^n \sim (\mu^{(t)})^n$ is the dataset for task t. Details of the dataset construction are provided in Section C.1. Let $\mathbf{S}$ denote the set of states $\{s^t_j\}$. Instantiating our framework, we learn a good representation by solving $\hat{\phi} = \arg\min_{\phi \in \Phi} \hat{L}(\phi)$, where

$$\begin{aligned}\hat{L}(\phi) &:= \frac{1}{T}\sum_{t=1}^{T} \min_{\pi \in \Pi^{\phi}} \frac{1}{n}\sum_{j=1}^{n} \ell(\pi(s^t_j), a^t_j) \\ &= \frac{1}{T}\sum_{t=1}^{T} \min_{\pi \in \Pi^{\phi}} \hat{\ell}^{\mathbf{x}^{(t)}}(\pi) \end{aligned} \quad (5)$$

where $\ell^{\mathbf{x}}$ is loss on samples $\mathbf{x} = \{(s_j, a_j)\}_{j=1}^n$ defined as $\ell^{\mathbf{x}}(\pi) = \frac{1}{n}\sum_{j=1}^n \ell(\pi(s_j), a_j)$.

Evaluating representation $\hat{\phi}$: A learned representation $\hat{\phi}$ is tested on a new task $\mu \sim \eta$ as follows: draw sam-

[2] We can easily extend the theory to non-stationary policies

ples $\mathbf{x} \sim \mu^n$ using trajectories from π^*_μ and solve $\pi^{\hat{\phi},\mathbf{x}} = \arg\min_{\pi \in \Pi^{\hat{\phi}}} \hat{\ell}^{\mathbf{x}}(\pi)$. Does $\pi^{\hat{\phi},\mathbf{x}}$ have expected cost $J_\mu(\pi^{\hat{\phi},\mathbf{x}})$ not much larger than $J_\mu(\pi^*_\mu)$? The following theorem answers this question. We make the following two assumptions to prove the theorem.

Assumption 5.1. *The expert policy π^*_μ is deterministic for every $\mu \in$ support(η).*

Assumption 5.2 (Policy realizability)**.** *There is a representation $\phi^* \in \Phi$ such that for every $\mu \in$ support(η), $\pi_\mu \in \Pi^{\phi^*}$ such that $\pi_\mu(s)_{\pi^*_\mu(s)}$[3] $\geq 1 - \gamma, \forall s \in \mathcal{S}$ for some $\gamma < 1/2$.*

The first assumption holds if π^*_μ is aiming to maximize some cost function. The second assumption is for representation learning to make sense: we need to assume the existence of a common representation ϕ^* that can approximate all expert policies and γ measures this expressiveness of Φ. Now we present our first main result about the performance of the learned representation on a new imitation learning task μ, whose performance is measure by the average cost J_μ.

Theorem 5.1. *Let $\hat{\phi} \in \arg\min_{\phi \in \Phi} \hat{L}(\phi)$. Under Assumptions 5.1,5.2, with probability $1 - \delta$ over the sampling of dataset $\mathbf{X}$, we have*

$$\mathop{\mathbb{E}}_{\mu \sim \eta} \mathop{\mathbb{E}}_{\mathbf{x} \sim \mu^n} J_\mu(\pi^{\hat{\phi},\mathbf{x}}) - \mathop{\mathbb{E}}_{\mu \sim \eta} J_\mu(\pi^*_\mu) \leq H^2(2\gamma + \epsilon_{gen})$$

where $\epsilon_{gen} = c\frac{G(\Phi(\mathbf{S}))}{T\sqrt{n}} + c'\frac{R\sqrt{K}}{\sqrt{n}} + c''\sqrt{\frac{\ln(4/\delta)}{T}}$, for some small constants c, c', c''.

To gain intuition for what the above result means, we give a PAC-style guarantee for the special case where the class of representation functions Φ is finite. This follows directly from the above theorem and the use of Massart's lemma.

Corollary 5.1. *In the same setting as Theorem 5.1, suppose Φ is finite. If number of tasks satisfies $T \geq c_1 \max\left\{\frac{H^4R^2\log(|\Phi|)}{\epsilon^2}, \frac{H^4\ln(4/\delta)}{\epsilon^2}\right\}$, and number of samples (expert trajectories) per task satisfies $n \geq c_2\frac{H^4R^2K}{\epsilon^2}$ for small constants c_1, c_2, then with probability $1 - \delta$,*

$$\mathop{\mathbb{E}}_{\mu \sim \eta} \mathop{\mathbb{E}}_{\mathbf{x} \sim \mu^n} J_\mu(\pi^{\hat{\phi},\mathbf{x}}) - \mathop{\mathbb{E}}_{\mu \sim \eta} J_\mu(\pi^*_\mu) \leq H^2\gamma + \epsilon$$

Discussion: The above bound says that as long as we have enough tasks to learn a representation from Φ and sufficient samples per task to learn a linear policy, the learned policy will have small average cost on a new task from η. The first term $H^2\gamma$ is small if the representation class Φ is expressive enough to approximate the expert policies (see Assumption 5.2). The results says that if we have access to data from $T = O\left(\frac{H^4R^2\log(|\Phi|)}{\epsilon^2}\right)$ tasks sampled from η, we can use them to learn a representation such that for a new task we only need $n = O\left(\frac{H^4R^2K}{\epsilon^2}\right)$ samples (expert demonstrations) to learn a linear policy with good performance. In contrast, without access to tasks, we would need $n = O\left(\max\left\{\frac{H^4R^2\log(|\Phi|)}{\epsilon^2}, \frac{H^4R^2K}{\epsilon^2}\right\}\right)$ samples from the task to learn a good policy $\pi \in \Pi$ from scratch. Thus if the complexity of the representation function class Φ is much more than number of actions ($\log(|\Phi|) \gg K$ in this case), then multi-task representation learning might be much more sample efficient[4]. Note that the dependence of sample complexity on H comes from the error propagation when going from ℓ^μ to J_μ; this is also observed in single task imitation learning (Ross et al., 2011; Sun et al., 2019).

We give a proof sketch for Theorem 5.1 below, while the full proof is deferred to Section A.

5.1. Proof sketch

The proof has two main steps. In the first step we bound the error due to use of samples. The policy $\pi^{\phi,\mathbf{x}}$ that is learned on samples $\mathbf{x} \sim \mu^n$ is evaluated on the distribution μ and the average loss incurred by representation ϕ across tasks is $\bar{L}(\phi) = \mathop{\mathbb{E}}_{\mu \sim \eta} \mathop{\mathbb{E}}_{\mathbf{x} \sim \mu^n} \ell^\mu(\pi^{\phi,\mathbf{x}})$.

On the other hand, if the learner had complete access to the distribution η and distributions μ for every task, then the loss minimizer would be $\phi^* = \arg\min_{\phi \in \Phi} L(\phi)$, where $L(\phi) := \mathop{\mathbb{E}}_{\mu \sim \eta} \min_{\pi \in \Pi^\phi} \ell^\mu(\pi)$. Using results from (Maurer et al., 2016), we can prove the following about $\hat{\phi}$

Lemma 5.2. *With probability $1 - \delta$ over the choice of $\mathbf{X}$, $\hat{\phi} \in \arg\min_{\phi \in \Phi} \hat{L}(\phi)$ satisfies*

$$\bar{L}(\hat{\phi}) \leq \min_{\phi \in \Phi} L(\phi) + c\frac{G(\Phi(\mathbf{S}))}{T\sqrt{n}} + c'\frac{R\sqrt{K}}{\sqrt{n}} + c''\sqrt{\frac{\ln(1/\delta)}{T}}$$

The proof of this lemma is provided in Section A.

The second step of the proof is connecting the loss $\bar{L}(\phi)$ and the average cost J_μ of the policies induced by ϕ for tasks $\mu \sim \eta$. This can obtained by using the connection between the surrogate 0-1 loss ℓ^μ and the cost J_μ that has been established in prior work (Ross et al., 2011; Syed & Schapire, 2010). The following lemma uses the result for deterministic expert policies from Ross et al. (2011).

Lemma 5.3. *Given a representation ϕ with $\bar{L}(\phi) \leq \epsilon$. Let $\mathbf{x} \sim \mu^n$ be samples for a new task $\mu \sim \eta$. Let $\pi^{\phi,\mathbf{x}}$ be the policy learned by behavioral cloning on the samples, then*

[3] We abuse notation and use $\pi^*_\mu(s)$ instead of $\arg\max_{a \in \mathcal{A}} \pi^*_\mu(s)_a$

[4] These statements are qualitative since we are comparing upper bounds.

under Assumption 5.1

$$\mathop{\mathbb{E}}_{\mu\sim\eta}\mathop{\mathbb{E}}_{\mathbf{x}\sim\mu^n} J_\mu(\pi^{\phi,\mathbf{x}}) - \mathop{\mathbb{E}}_{\mu\sim\eta} J_\mu(\pi^*_\mu) \le H^2\epsilon$$

This suggests that representations with small $\bar{L}$ do well on the imitation learning tasks. A simple implication of Assumption 5.2 that $\min_{\phi\in\Phi} L(\phi) \le L(\phi^*) \le \gamma$, along with the above two lemmas completes the proof.

6. Representation Learning for Observations Alone Setting

Now we consider the setting where we cannot observe experts' actions but only their states. As in Sun et al. (2019), we also solve a problem at each level; consider a level $h \in [H]$.

Choice of ℓ_h^μ: Let $\boldsymbol{\pi}^*_\mu = \{\pi^*_{1,\mu}, \dots, \pi^*_{H,\mu}\}$ be the sequence of expert policies (possibly stochastic) at different levels for the task μ. Let $\nu^*_{h,\mu}$ be the distribution induced on the states at level h by the expert policy $\boldsymbol{\pi}^*_\mu$. The goal in imitation learning with observations alone (Sun et al., 2019) is to learn a policy $\boldsymbol{\pi} = (\pi_1, \dots, \pi_H)$ that matches the distributions ν^π_h with ν^*_h for every h, w.r.t. a discriminator class $\mathcal{G}$[5] that contains the true value functions $V^*_1, \dots, V^*_H$ and is *approximately* closed under the Bellman operator of $\boldsymbol{\pi}^*$. Instead, in this work we learn $\boldsymbol{\pi}$ that matches the distributions $\pi_h \cdot \nu^*_h$[6] and ν^*_{h+1} for every h w.r.t. to a class $\mathcal{G} \subseteq \{g : \mathcal{S} \to \mathbb{R}, |g|_\infty \le 1\}$ that contains the value functions and has a stronger Bellman operator closure property. For every task μ, ℓ^μ_h is defined as

$$\ell^\mu_h(\pi) = \max_{g\in\mathcal{G}}\Big[\mathop{\mathbb{E}}_{s\sim\nu^*_{h,\mu}}\mathop{\mathbb{E}}_{\substack{a\sim\pi(s)\\ \tilde{s}\sim P_{s,a}}} g(\tilde{s}) - \mathop{\mathbb{E}}_{\bar{s}\sim\nu^*_{h+1,\mu}} g(\bar{s})\Big] \tag{6}$$

$$= \max_{g\in\mathcal{G}}\Big[\mathop{\mathbb{E}}_{s\sim\nu^*_{h,\mu}}\mathop{\mathbb{E}}_{\substack{a\sim\mathcal{U}(\mathcal{A})\\ \tilde{s}\sim P_{s,a}}} K\pi(a|s)g(\tilde{s}) - \mathop{\mathbb{E}}_{\bar{s}\sim\nu^*_{h+1,\mu}} g(\bar{s})\Big]$$

where we rewrite ℓ^μ_h by importance sampling in the second equation; this will be useful to get an empirical estimate. While our definition of ℓ^μ_h differs slightly from the one used in Sun et al. (2019), using similar techniques, we will show that small values for $\ell^\mu_h(\pi_h)$ for every $h \in [H]$ will ensure that the policy $\boldsymbol{\pi} = (\pi_1, \dots, \pi_H)$ will have expected cost $J_\mu(\boldsymbol{\pi})$ close to $J_\mu(\boldsymbol{\pi}^*_\mu)$. We abuse notation, and for a task μ we denote $\mu = (\mu_1, \dots, \mu_H)$ where μ_h is the distribution of $(s, a, \tilde{s}, \bar{s})$ used in ℓ^μ_h; thus $(s, a, \tilde{s}, \bar{s}) \sim \mu_h$ is equivalent to $s \sim \nu^*_{h,\mu}, a \sim \mathcal{U}(\mathcal{A}), \tilde{s} \sim P_{s,a}, \bar{s} \sim \nu^*_{h+1,\mu}$.

Learning ϕ_h from samples: We assume, 1) access to $2n$ expert trajectories for T independent *train* tasks, 2) ability to reset the environment at any state s and sample from the transition $P(\cdot|s, a)$ for any $a \in \mathcal{A}$. The second condition is satisfied in many problems equipped with simulators. Using the sampled trajectories for the T tasks $\{\mu^{(1)}, \dots, \mu^{(T)}\}$ and doing some interaction with environment, we get the following dataset $\mathbf{X} = \{\mathbf{X}_1, \dots, \mathbf{X}_H\}$ where $\mathbf{X}_h$ is the dataset for level h. Specifically, $\mathbf{X}_h = \{\mathbf{x}^{(1)}_h, \dots, \mathbf{x}^{(T)}_H\}$ where $\mathbf{x}^{(i)}_h = \{(s^i_j, a^i_j, \tilde{s}^i_j, \bar{s}^i_j)\}^n_{j=1} \sim (\mu^{(i)})^n$ is the dataset for task i at level h. Additionally we denote $\mathbf{S}_h = \{s^i_j\}^{T,n}_{i=1,j=1}$ to be all the s-states in $\mathbf{X}_h$, $\tilde{\mathbf{S}}_h$ and $\bar{\mathbf{S}}_h$ are similarly defined as the collections of all the $\tilde{s}$-states and $\bar{s}$-states respectively. Details about how this dataset is constructed from expert trajectories and interactions with environment is provided in Section C.2. We learn the representation $\hat{\phi}_h = \arg\min_{\phi\in\Phi} \hat{L}_h(\phi)$, where

$$\hat{L}_h(\phi) = \frac{1}{T}\sum_{i=1}^{T}\min_{\pi\in\Pi^\phi}\max_{g\in\mathcal{G}}\frac{1}{n}\sum_{j=1}^{n}[K\pi(a^i_j|s^i_j)g(\tilde{s}^i_j) - g(\bar{s}^i_j)]$$

$$= \frac{1}{T}\sum_{i=1}^{T}\min_{\pi\in\Pi^\phi}\hat{\ell}^{\mathbf{x}^{(i)}}_h(\pi) \tag{7}$$

where for dataset $\mathbf{x} = \{(s_j, a_j, \tilde{s}_j, \bar{s}_j)\}^n_{j=1}$, $\hat{\ell}^{\mathbf{x}}_h(\pi) := \max_{g\in\mathcal{G}}\frac{1}{n}\sum_{j=1}^{n}[K\pi(a_j|s_j)g(\tilde{s}_j) - g(\bar{s}_j)]$. Note that because of the max operator over the class $\mathcal{G}$, $\hat{\ell}^{\mathbf{x}}_h$ is not an unbiased estimator of ℓ^μ_h when $\mathbf{x} \sim \mu^n_h$. However we can still show generalization bounds.

Evaluating representations $\hat{\phi}_1, \dots, \hat{\phi}_H$: Learned representations are tested on a new task $\mu \sim \eta$ as follows: get samples $\mathbf{x} = (\mathbf{x}_1, \dots, \mathbf{x}_H)$[7] for all levels using trajectories from $\boldsymbol{\pi}^*_\mu$, where $\mathbf{x}_h \sim \mu^n_h$. For each level h, learn $\pi^{\hat{\phi}_h,\mathbf{x}_h} = \arg\min_{\pi\in\Pi^{\hat{\phi}}}\hat{\ell}^{\mathbf{x}_h}_h(\pi)$ and consider the policy $\boldsymbol{\pi}^{\hat{\phi},\mathbf{x}} = (\pi^{\hat{\phi}_1,\mathbf{x}_1}, \dots, \pi^{\hat{\phi}_H,\mathbf{x}_H})$. Before presenting the guarantee for $\pi^{\hat{\phi},\mathbf{x}}$, we introduce a notion of *Bellman error* that will show up in our results. For a policy $\boldsymbol{\pi} = (\pi_1, \dots, \pi_H)$ and an expert policy $\boldsymbol{\pi}^* = (\pi^*_1, \dots, \pi^*_H)$, we define the inherent Bellman error

$$\epsilon^{\boldsymbol{\pi}}_{be} := \max_{h\in[H]}\max_{g\in\mathcal{G}}\min_{g'\in\mathcal{G}}\mathop{\mathbb{E}}_{s\sim(\nu^*_h+\nu^{\boldsymbol{\pi}}_h)/2}[|g'(s) - (\Gamma^{\boldsymbol{\pi}}_h g)(s)|] \tag{8}$$

We make the following two assumptions for the subsequent theorem. These are standard assumptions in theoretical reinforcement learning literature.

Assumption 6.1 (Value function realizability)**.** *$V^*_{h,\mu} \in \mathcal{G}$ for every $h \in [H]$, $\mu \in$ support(η).*

[5] If $\mathcal{G}$ contains all bounded functions, then it reduces to minimizing TV between ν^π_h and ν^*_h.

[6] The sampling $s \sim \pi_h \cdot \nu^*_h$ is defined as sampling $s' \sim \nu^*_h, a \sim \pi_h(s'), s \sim P_{s',a}$.

[7] Note that we do not need the datasets $\mathbf{x}_h$ at different levels to be independent of each other

Assumption 6.2 (Policy realizability)**.** *There are representations $\phi_1^*, \ldots, \phi_H^* \in \Phi$ such that $\pi_{h,\mu}^* \in \Pi^{\phi_h^*}$ for every $h \in [H]$, $\mu \in$ support(η).*

Now we present our main theorem for the observation-alone setting.

Theorem 6.1. *Let $\hat{\phi}_h \in \arg\min_{\phi \in \Phi} \hat{L}_h(\phi)$. Under Assumptions 6.1,6.2, with probability $1-\delta$ over sampling of $\mathbf{X} = (\mathbf{X}_1, \ldots, \mathbf{X}_H)$, we have*

$$\mathop{\mathbb{E}}_{\mu\sim\eta}\mathop{\mathbb{E}}_{\mathbf{x}} J(\boldsymbol{\pi}^{\hat{\phi},\mathbf{x}}) - \mathop{\mathbb{E}}_{\mu\sim\eta} J(\boldsymbol{\pi}_\mu^*) \leq \sum_{h=1}^{H}(2H-2h+1)\epsilon_{gen,h} + O(H^2)\epsilon_{be}^{\hat{\phi}}$$

where $\epsilon_{be}^{\hat{\phi}} = \mathop{\mathbb{E}}_{\mu\sim\eta}\mathop{\mathbb{E}}_{\mathbf{x}}[\epsilon_{be}^{\boldsymbol{\pi}^{\hat{\phi},\mathbf{x}}}]$ is the average inherent Bellman error and

$$\epsilon_{gen,h} = c_1\frac{KG(\Phi(\mathbf{S}_h))}{T\sqrt{n}} + c_2\frac{RK\sqrt{K}}{\sqrt{n}} + c_3\sqrt{\frac{\ln(H/\delta)}{T}} + c_4\mathop{\mathbb{E}}_{\mu\sim\eta}\mathop{\mathbb{E}}_{\mathbf{x}\sim\mu^n}\left[\frac{KG(\mathcal{G}(\tilde{\mathbf{s}}_h))}{n} + \frac{G(\mathcal{G}(\bar{\mathbf{s}}_h))}{n}\right]$$

We again give a PAC-style guarantee for the special case where the class of representation functions Φ and value function class $\mathcal{G}$ are finite. It follows from the above theorem and Massart's lemma.

Corollary 6.1. *In the setting of Theorem 6.1, suppose $\Phi, \mathcal{G}$ are finite. If number of tasks satisfies $T \geq c_1 \max\left\{\frac{H^4R^2K^2\log(|\Phi|)}{\epsilon^2}, \frac{H^4\ln(H/\delta)}{\epsilon^2}\right\}$, and number of samples (trajectories) per task satisfies $n \geq c_2 \max\left\{\frac{H^4K^2\log(|\mathcal{G}|)}{\epsilon^2}, \frac{H^4R^2K^3}{\epsilon^2}\right\}$ for small constants c_1, c_2, then with probability $1-\delta$,*

$$\mathop{\mathbb{E}}_{\mu\sim\eta}\mathop{\mathbb{E}}_{\mathbf{x}} J(\boldsymbol{\pi}^{\hat{\phi},\mathbf{x}}) - \mathop{\mathbb{E}}_{\mu\sim\eta} J(\boldsymbol{\pi}_\mu^*) \leq O(H^2)\epsilon_{be}^{\hat{\phi}} + \epsilon.$$

Discussion: As in the previous section, the number of samples required for a new task after learning a representation is independent of the class Φ but depends only on the value function class $\mathcal{G}$ and number of actions. Thus representation learning is very useful when the class Φ is much more complicated than $\mathcal{G}$, i.e. $R^2\log(|\Phi|) \gg \max\{\log(|\mathcal{G}|), R^2K\}$. In the above bounds, $\epsilon_{be}^{\hat{\phi}}$ is a Bellman error term. This type of error terms occur commonly in the analysis of policy iteration type algorithms (Munos, 2005; Munos & Szepesvári, 2008). We remark that unlike in Sun et al. (2019), our Bellman error is based on the Bellman operator of the learned policy rather than the optimal policy. Le et al. (2019) used a similar notion that they call *inherent Bellman evaluation error*.

The proof of Theorem 6.1 follows a similar outline to that of behavioral cloning. However we cannot use results from (Maurer et al., 2016) directly since we are solving a min-max game for each task. We provide the proof in Section B.

7. Experiments

In this section we present our experimental results. These experiments have two aims:

1. Verify our theory that representation learning can reduce the sample complexity in the new imitation learning task.
2. Test the power of representations learned via our framework in a broader context. We wish to see if the learned representation is useful beyond imitation learning and can be used to learn a policy in the RL setting.

Since our goal of the experiment is to demonstrate the advantage of representation learning, we only consider the standard baseline where for a task we learn a policy π from the class Π from scratch (without learning a representation first using other tasks).

We conduct our experiments on two environments. NoisyCombinationLock is a variant of the standard CombinationLock environment (Kakade et al., 2003), we add additional noisy features to confuse the agent. Different tasks involve different combinations for the lock. SwimmerVelocity is a modifed environment the Swimmer environment from OpenAI gym (Brockman et al., 2016) with Mujoco simulator (Todorov et al., 2012), and this environment is similar to the one used in (Finn et al., 2017a). The goal in SwimmerVelocity is to move at a target velocity (speed and direction) and the various tasks differ in target velocities. See Section D for more details about these two environments.

7.1. Verification of Theory

We first present our experimental results to verify our theory.

Representation learning for Behavioral Cloning We first test our theory on representation learning for behavioral cloning. We learn the representation using Equation 5 on the first T tasks. The specification of policy class and other experiment details are in Section D.

The first plot in Figure 1 shows results on the NoisyCombinationLock environment. We observe that in NoisyCombinationLock, even one expert can help and more experts will always improve the average return.

The second plot in Figure 1 shows results on the SwimmerVelocity environment. Again, more experts always help. Furthermore, we observe an interesting phenomenon. When the number of experts is small (2 or 4), the baseline method can outperform policies trained using representation learning, though the baseline method requires more samples to

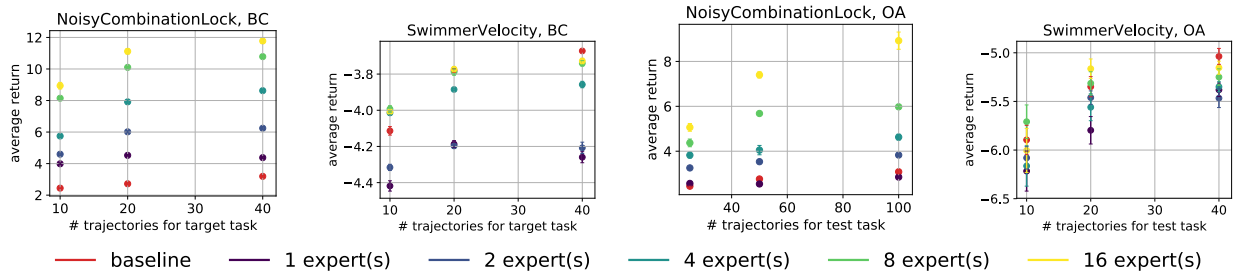

Figure 1. Experiments for verifying our theory. From left to right: Representation learning for behavioral cloning on NoisyCombinationLock, representation learning for behavioral cloning on SwimmerVelocity, representation learning for observations alone setting on NoisyCombinationLock, representation learning for observations alone setting on SwimmerVelocity, We compare imitation learning based on learned representation using 1 - 16 experts to the baseline method (without representation learning). The error bars are calculated using 5 seeds and indicate one standard deviation.

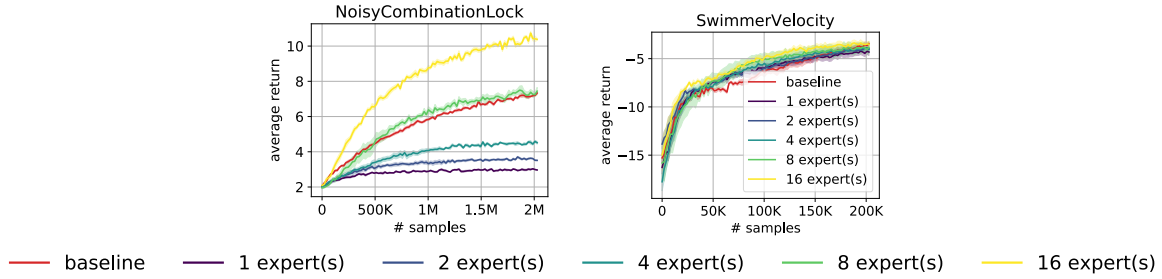

Figure 2. Experiments on Policy Optimization with representation trained by imitation learning. Left: Results on NoisyCombinationLock. Right: Results on SwimmerVelocity. We compare policy optimization based on learned representation using 1 - 16 experts to the baseline method (without representation learning).

achieve this. This behavior is actually expected according to our theory. When the number of experts is small, we may learn a sub-optimal representation and because we fix this representation for training the policy, more samples for the test task cannot make this policy better, whereas more samples always make the baseline method better.

Representation Learning for Observations Alone Setting We next verify our theory for the observations alone setting. We learn the representation using Equation 7 on the first T tasks. Again, the specification of policy class and other experiment details are in Section D.

The results for NoisyCombinationLock and SwimmerVelocity are shown in the third and the fourth plots in Figure 1, respectively. We observe similar phenomenon as the first and the second plot. Increasing the number of experts always help and baseline method can outperform policies trained using representation learning when the number of trajectories for the test task is large.

We remark that comparing with the behavioral cloning setting, the observations alone setting often has smaller return. We suspect the reason is that Equation 7 considers the worst case g in $\mathcal{G}$, thus it prefers pessimistic policies. Also this setting does not have access to the experts actions as opposed to the behavioral cloning setting.

7.2. Policy optimization with representations trained by imitation learning

We test whether the learned representation via imitation learning is useful for the target *reinforcement learning* problem. We use Equation 5 to learn representations and we use a proximal policy optimization method (Schulman et al., 2017) to learn a linear policy over the learned representation. See Section D for details.

The results are reported in Figure 2 and are very encouraging. For both NoisyCombinationLock and SwimmerVelocity environments, we observe that when the number of experts to learn the representation is small, the baseline method enjoys better performance than the policies trained using representation learning. On the other hand, as the number of experts increases, the policy trained using representation learning can outperform the baseline, sometime significantly. This experiment suggests that representations trained via imitation learning can be useful *beyond imitation learning*, especially when the target task has few samples.

8. Conclusion

The current paper proposes a bi-level optimization framework to formulate and analyze representation learning for imitation learning using multiple demonstrators. Theoretical guarantees are provided to justify the statistical benefit of representation learning. Some preliminary experiments verify the effectiveness of the proposed framework. In particular, in experiments, we find the representation learned via imitation learning is also useful for policy optimization in the reinforcement learning setting. We believe it is an interesting theoretical question to explain this phenomenon. Additionally, extending this bi-level optimization framework to incorporate methods beyond imitation learning is an interesting future direction. Finally, while we fix the learned representation for a new task, once could instead also fine-tune the representation given samples for a new task and a theoretical analysis of this would be of interest.

9. Acknowledgments

We thank Wen Sun for useful discussions. Simon S. Du is supported by NSF grant DMS-1638352 and the Infosys Membership. Sanjeev Arora, Yuping Luo and Nikunj Saunshi are supported by NSF, ONR, Simons Foundation, Schmidt Foundation, Amazon Research, DARPA and SRC. Sham Kakade acknowledges funding from the Washington Research Foundation for Innovation in Data-intensive Discovery, the ONR award N00014-18-1-2247, NSF Award CCF-1703574, and the NSF Award CCF-1637360.

References

Arora, S., Khandeparkar, H., Khodak, M., Plevrakis, O., and Saunshi, N. A theoretical analysis of contrastive unsupervised representation learning. In *Proceedings of the 36th International Conference on Machine Learning*, 2019.

Bartlett, P. L. and Mendelson, S. Rademacher and gaussian complexities: Risk bounds and structural results. *J. Mach. Learn. Res.*, 2003.

Baxter, J. A model of inductive bias learning. *J. Artif. Int. Res.*, 2000.

Bengio, Y., Courville, A., and Vincent, P. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 08 2013.

Brockman, G., Cheung, V., Pettersson, L., Schneider, J., Schulman, J., Tang, J., and Zaremba, W. Openai gym, 2016.

Bullins, B., Hazan, E., Kalai, A., and Livni, R. Generalize across tasks: Efficient algorithms for linear representation learning. In *Proceedings of the 30th International Conference on Algorithmic Learning Theory*, 2019.

Chang, K.-W., Krishnamurthy, A., Agarwal, A., Daumé, III, H., and Langford, J. Learning to search better than your teacher. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*, ICML'15. JMLR.org, 2015.

Daumé, Iii, H., Langford, J., and Marcu, D. Search-based structured prediction. *Mach. Learn.*, 2009.

Denevi, G., Ciliberto, C., Stamos, D., and Pontil, M. Incremental learning-to-learn with statistical guarantees. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, 2018.

Denevi, G., Ciliberto, C., Grazzi, R., and Pontil, M. Learning-to-learn stochastic gradient descent with biased regularization. In *Proceedings of the 36th International Conference on Machine Learning*, 2019.

Dhariwal, P., Hesse, C., Klimov, O., Nichol, A., Plappert, M., Radford, A., Schulman, J., Sidor, S., Wu, Y., and Zhokhov, P. Openai baselines. https://github.com/openai/baselines, 2017.

Duan, Y., Andrychowicz, M., Stadie, B., Jonathan Ho, O., Schneider, J., Sutskever, I., Abbeel, P., and Zaremba, W. One-shot imitation learning. In *Advances in Neural Information Processing Systems 30*. 2017.

Edwards, A. D., Sahni, H., Schroecker, Y., and Isbell, C. L. Imitating latent policies from observation. *arXiv preprint arXiv:1805.07914*, 2018.

Finn, C., Abbeel, P., and Levine, S. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning*, 2017a.

Finn, C., Yu, T., Zhang, T., Abbeel, P., and Levine, S. One-shot visual imitation learning via meta-learning. 09 2017b.

Finn, C., Rajeswaran, A., Kakade, S., and Levine, S. Online meta-learning. In *Proceedings of the 36th International Conference on Machine Learning*, 2019.

Ho, J. and Ermon, S. Generative adversarial imitation learning. In *NIPS*, 2016.

James, S., Bloesch, M., and Davison, A. Task-embedded control networks for few-shot imitation learning. 10 2018.

Kakade, S. M. et al. *On the sample complexity of reinforcement learning*. PhD thesis, University of London London, England, 2003.

Khodak, M., Balcan, M.-F., and Talwalkar, A. Adaptive gradient-based meta-learning methods. *arXiv preprint arXiv:1906.02717*, 2019.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Le, H., Voloshin, C., and Yue, Y. Batch policy learning under constraints. In *Proceedings of the 36th International Conference on Machine Learning*, pp. 3703–3712, 2019.

Maurer, A. Transfer bounds for linear feature learning. *Machine Learning*, 2009.

Maurer, A., Pontil, M., and Romera-Paredes, B. The benefit of multitask representation learning. *The Journal of Machine Learning Research*, 17(1):2853–2884, 2016.

Munos, R. Error bounds for approximate value iteration. In *Proceedings of the 20th National Conference on Artificial Intelligence - Volume 2*, AAAI'05. AAAI Press, 2005.

Munos, R. and Szepesvári, C. Finite-time bounds for fitted value iteration. *J. Mach. Learn. Res.*, 2008.

Pan, Y., Cheng, C.-A., Saigol, K., Lee, K., Yan, X., Theodorou, E., and Boots, B. Agile autonomous driving using end-to-end deep imitation learning. In *Proceedings of Robotics: Science and Systems*, 2018.

Pomerleau, D. A. Efficient training of artificial neural networks for autonomous navigation. *Neural Computation*, 3, 1991.

Raghu, A., Raghu, M., Bengio, S., and Vinyals, O. Rapid learning or feature reuse? towards understanding the effectiveness of maml. *arXiv preprint arXiv:1909.09157*, 2019.

Rajeswaran, A., Finn, C., Kakade, S., and Levine, S. Meta-learning with implicit gradients. *arXiv preprint arXiv:1906.02717*, 2019.

Ross, S. and Bagnell, D. Efficient reductions for imitation learning. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pp. 661–668, 2010.

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

Ross, S., Gordon, G., and Bagnell, D. 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*, 2011.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Sun, J., Qu, Q., and Wright, J. Complete dictionary recovery over the sphere I: Overview and the geometric picture. *IEEE Transactions on Information Theory*, 63(2):853–884, 2017.

Sun, W., Bagnell, J. A., and Boots, B. Truncated horizon policy search: Combining reinforcement learning and imitation learning. *arXiv preprint arXiv:1805.11240*, 2018.

Sun, W., Vemula, A., Boots, B., and Bagnell, J. A. Provably efficient imitation learning from observation alone. *arXiv preprint arXiv:1905.10948*, 2019.

Syed, U. and Schapire, R. E. A reduction from apprenticeship learning to classification. In *Advances in Neural Information Processing Systems 23*, pp. 2253–2261. 2010.

Todorov, E., Erez, T., and Tassa, Y. Mujoco: A physics engine for model-based control. pp. 5026–5033. IEEE, 2012. URL http://dblp.uni-trier.de/db/conf/iros/iros2012.html#TodorovET12.

Torabi, F., Warnell, G., and Stone, P. Behavioral cloning from observation. In *IJCAI*, 2018.

Wu, J., Lu, E., Kohli, P., Freeman, B., and Tenenbaum, J. B. Learning to see physics via visual de-animation. In *NIPS*, 2017.