On-Policy Imitation Learning from an Improving Supervisor

Ashwin Balakrishna [*1] **Brijen Thananjeyan** [*1] **Jonathan Lee** [1] **Arsh Zahed** [1] **Felix Li** [1]
Joseph E. Gonzalez [1] **Ken Goldberg** [1]

Abstract

Most on-policy imitation algorithms, such as DAgger, are designed for learning with a fixed supervisor. However, there are many settings in which the supervisor improves during policy learning, such as when the supervisor is a human performing a novel task or an improving algorithmic controller. We consider learning from an "improving supervisor" and derive a bound on the static-regret of online gradient descent when a converging supervisor policy is used. We present an on-policy imitation learning algorithm, Follow the Improving Teacher (FIT), which uses a deep model-based reinforcement learning (deep MBRL) algorithm to provide the sample complexity benefits of model-based methods but enable faster training and evaluation via distillation into a reactive controller. We evaluate FIT with experiments on the Reacher and Pusher MuJoCo domains using the deep MBRL algorithm, PETS, as the improving supervisor. To the best of our knowledge, this work is the first to formally consider the setting of an improving supervisor in on-policy imitation learning.

1. Introduction

In on-policy imitation learning, a policy is iteratively trained to match the behavior of a supervisor on a particular task on the distribution of the learned policy. In algorithms such as DAgger (Ross et al., 2011a), the supervisor serves as a labeler, providing feedback on the appropriate controls for states visited by the learner. Ross et al. (2011a) show that DAgger can be interpreted as a no-regret algorithm in the online-learning setting, and provides vanishing regret guarantees when the policy update step via Follow The Leader (FTL) has vanishing regret (Ross et al., 2011a; Kakade & Tewari, 2009).

Prior work focuses on imitation learning algorithms with a fixed supervisor (Ross et al., 2011a; Sun et al., 2017; Lee et al., 2019; Cheng & Boots, 2018). However, in this work, we consider a convergent sequence of supervisors. This context is motivated by practical scenarios in which the supervisor may improve its task performance substantially as time progresses, e.g., as a human supervisor learns how to play a game they have never played or teleoperate a robot with unfamiliar controls.

We investigate how initially suboptimal labeling feedback affects the incurred static regret of the learned policy. This is particularly relevant to long time horizon tasks, in which a large-scale system is designed to improve over time on a difficult task using human experience as feedback. In this work, we show that results are not significantly affected when the supervisor is initially suboptimal, as long as it converges to the desired policy.

Learning from improving supervisors also has applications to deep model-based reinforcement learning, which has attracted interest due to the improved sample-efficiency compared to model-free methods (Chua et al., 2018). Recent model-based RL algorithms for continuous-control domains represent system dynamics with a deep neural network, which is updated on-policy, and use model-predictive control (MPC) to generate controls (Chua et al., 2018; Nagabandi et al., 2018). However, generating controls for dynamics models represented by deep neural networks often involves significant online computation, making it infeasible to collect high-frequency policy rollouts from the model-based controller. This significantly slows down both training, which requires policy rollouts for policy evaluation, and evaluation at test-time, making direct application of these techniques difficult in many robotic tasks. We focus on this setting in this work.

Motivated by the idea of learning from an improving supervisor, we present an on-policy imitation learning algorithm to train a model-based deep reinforcement learning agent using off-policy data from a model-free learner policy. The model-based supervisor is used to generate labels, which are then used to update the learner. This enables fast policy eval-

[*] Equal contribution [1] Department of EECS, University of California, Berkeley. Correspondence to: Ashwin Balakrishna <ashwin_balakrishna@eecs.berkeley.edu>, Brijen Thananjeyan <brijen@eecs.berkeley.edu>.

Real-World Sequential Decision Making Workshop at ICML 2019, Copyright 2019 by the author(s).

uation since rollouts are collected from a model-free policy, so the deep model-based controller can relabel visited states in a rollout in parallel at the end of a rollout rather than serially. This combines the sample complexity advantages of deep model-based methods with the fast policy evaluation of model-free RL.

The contributions of this work are:

1. New setting for on-policy imitation learning and sublinear regret guarantees if online gradient descent (OGD) is used to learn from an improving supervisor.
2. A new algorithm, FIT, which involves training a model-free policy to track an improving deep model-based RL algorithmic supervisor.
3. Experimental data suggesting that FIT can achieve final return within 16 % of the method proposed in (Chua et al., 2018) on both the Reacher and Pusher MuJoCo domains while utilizing fully off-policy data from a model-free policy, suggesting the potential for significant speedups in learning and policy evaluation.

2. Related Work

Imitation learning is an effective method for learning complex behaviors efficiently from the demonstration data of a supervisor. In this section, we review related work in online policy learning via imitation and deep model-based reinforcement learning. We also briefly discuss past work in supervised and active learning involving learning from stochastic and multiple supervisors.

2.1. Learning from Stochastic Supervisors

Past work in supervised learning involving learning from noisy labels (Natarajan et al., 2013; Khetan et al., 2018) typically has focused mainly on classification. Common settings are cases where labels either are noise-injected versions of oracle labels (Natarajan et al., 2013) or labels come from a variety of different supervisors (Khetan et al., 2018). There has also been interest in intelligently aggregating the results of multiple labelers for each instance in a dataset (Raykar et al., 2009; Richardson & Domingos, 2003; Laskey et al., 2016a). Finally, there has also been work in active learning to determine which of a set of supervisors, of varying quality, to query for labels (Zhang & Chaudhuri, 2015; Yan et al., 2012). Distinct from these works, we focus specifically on the imitation learning setting and provide analysis when a supervisor is improving over time.

2.2. On-Policy Imitation Learning

Online imitation learning algorithms that directly learn reactive policies from a supervisor were recently popularized with DAgger in (Ross et al., 2011b). DAgger is an iterative approach that improves by deploying the learner's current policy and receiving supervisor feedback. It was shown that this approach achieves significant performance gains in both theory and practice over analogous offline methods, e.g. (Bagnell, 2015; Pomerleau, 1989). Since DAgger, a number of practical extensions have been proposed to, for example, ensure safety (Menda et al., 2017), smooth controllers (Le et al., 2016), and relieve burden on human supervisors (Laskey et al., 2016b). Furthermore, online methods have been applied with both human (Laskey et al., 2017) and algorithmic supervisors (Pan et al., 2018) such as MPC. These works typically assume that a single, fixed supervisor is the guiding policy. We propose a setting in which the supervisor improves over time, which is common when learning from human demonstrators or when distilling an expensive, iteratively improving controller into a policy that can be efficiently executed in practice.

Since the introduction of DAgger, theoretical analyses of the online imitation learning algorithms (Ross et al., 2011b; Sun et al., 2017) have often been embedded in the online learning framework (Zinkevich, 2003). Recently, convergence results and stronger guarantees on regret metrics such as dynamic regret have been shown for a fixed supervisor (Cheng & Boots, 2018; Lee et al., 2019). We present an analysis of on-policy imitation learning from a convergent sequence of supervisors. Specifically, we show that OGD can achieve comparable guarantees in the improving supervisor setting.

2.3. Deep Model-Based Reinforcement Learning

Deep model-based reinforcement learning has seen significant interest due to the improvement in sample complexity when compared to model-free methods (Deisenroth & Rasmussen, 2011; Levine et al., 2015; Nagabandi et al., 2018; Chua et al., 2018). Recently, Nagabandi et al. (2018) and Chua et al. (2018) showed that using neural network dynamics models to perform MPC gives comparable asymptotic performance to model-free reinforcement learning methods while maintaining the sample complexity gains of model-based methods. However, solving the MPC objective over neural network dynamics models often requires expensive, derivative-free, sampling-based approaches such as the Cross-Entropy Method (CEM), significantly slowing down policy evaluation. In this work, we investigate how distilling MPC into a model-free policy by treating it as an improving supervisor can be used to accelerate policy evaluation and thus potentially speed up both training and evaluation of these methods. The closest related work in this regard is (Kahn et al., 2017). Here, an MPC policy is trained with true state information, but constrained to be close to a model-free policy which only has access to observations. However, Kahn et al. (2017) assume known dynamics, and thus the MPC supervisor does not improve with time.

3. Improving Supervisor Framework and Analysis

On-policy imitation learning involves executing a policy in the environment, and then soliciting feedback from a supervisor on the visited states. This is in contrast to off-policy imitation learning methods, such as behavior cloning, in which policy learning is performed entirely on states from the supervisor's state distribution rather than that of the learner. Here we outline a theoretical framework in which to study on-policy imitation learning with an improving supervisor and provide an analysis of the static regret of OGD in this setting.

3.1. Definitions

1. **Supervisor:** Consider a sequence of N supervisors (labelers), $(\psi_i)_{i=1}^N$, where ψ_i is any function on $\mathbb{R}^S \to \mathbb{R}^A$, $\mathcal{S}$ and $\mathcal{A}$ are the sets of allowed states and actions, and S and A are the dimensionality of these sets respectively. Supervisor ψ_i provides labels for imitation learning policy updates at iteration i.
2. **Learner:** The learner is represented at iteration i by a parameterized policy π_{θ_i} on $\mathbb{R}^S \to \mathbb{R}^A$ where π_{θ_i} is differentiable in the policy parameter $\theta_i \in \mathcal{K}$. $\Pi_{\mathcal{K}}$ denotes the Euclidean projection operator onto $\mathcal{K}$.
3. **Loss:** We specifically consider losses of the following form:

$$l_i(\pi_{\theta_i}, \psi) = \mathbb{E}_{s \in \{X_i\}} \|\pi_{\theta_i}(s) - \psi(s)\|^2$$

where $\|\cdot\|$ is the 2-norm and $\{X_i\}$ is the set of states in the minibatch processed at iteration i.

4. **Regret:** We analyze the regret of FIT with respect to the best policy in hindsight that has labels from the final supervisor ψ_N for the whole dataset. Note, however, that during learning, labels are provided not by ψ_N, but by supervisor ψ_i at iteration i. This results in a more difficult regret metric than is typically considered in static regret analysis for on-policy imitation learning since labels are provided by the improving supervisor but regret is evaluated with respect to the best policy given labels from the final supervisor.

$$\text{Regret}_N = \sum_{i=1}^{N} l_i(\pi_{\theta_i}, \psi_N) - \sum_{i=1}^{N} l_i(\pi_{\theta^\star}, \psi_N)$$

$$\text{where } \theta^\star = \arg\min_{\theta \in \mathcal{K}} \sum_{i=1}^{N} l_i(\pi_\theta, \psi_N)$$

3.2. Assumptions

In our regret analysis we adopt the following standard assumptions.

1. **Strongly convex losses:** $l_i(\pi_{\theta_i}, \psi)$ is strongly convex with respect to θ with parameter $\alpha \in \mathbb{R}^+$. Precisely, we assume that

$$l(\pi_{\theta_2}, \psi) \geq l(\pi_{\theta_1}, \psi) + \nabla_\theta l(\pi_{\theta_1}, \psi)^T(\theta_2 - \theta_1) + \frac{\alpha}{2}\|\theta_2 - \theta_1\|_2^2 \;\forall\, \theta_1, \theta_2 \in \mathcal{K}$$

2. **Bounded parameter space diameter:** $\|\theta_i - \theta_j\| \leq D \;\forall\, \theta_i, \theta_j \in \mathcal{K}$ where $D \in \mathbb{R}^+$.
3. **Bounded action space:** The diameter of the action space $\mathcal{A}$ of the policy is bounded, specifically that

$$\|a_1 - a_2\|_2 \leq \delta \quad \forall\, a_1, a_2 \in \mathcal{A}$$

where $\delta \in \mathbb{R}^+$.

4. **Bounded operator norm of policy Jacobian:** $\|\nabla_\theta \pi_{\theta_i}(s)\| \leq G$ for all $s \in \mathcal{S}$ where $G \in \mathbb{R}^+$ and $\|\cdot\|$ is a subadditive operator norm. Note that this also implies that the loss function gradients are bounded since

$$\|\nabla_\theta l(\pi_{\theta_i}, \psi)\| = \\ \|\mathbb{E}_{s\in\{X_i\}}\left[2(\nabla_\theta \pi_{\theta_i}(s))^T\left(\pi_{\theta_i}(s) - \psi_i(s)\right)\right]\| \leq \\ \mathbb{E}_{s\in\{X_i\}}\left[\|2(\nabla_\theta \pi_{\theta_i}(s))^T\left(\pi_{\theta_i}(s) - \psi_i(s)\right)\|\right]$$

by convexity of norms $\|\cdot\|$ and Jensen's inequality.

Then, we have that

$$\|(\nabla_\theta \pi_{\theta_i}(s))^T(\pi_{\theta_i}(s) - \psi_i(s))\| \leq \\ \|\nabla_\theta \pi_{\theta_i}(s)\|\|\pi_{\theta_i}(s) - \psi_i(s)\| \leq G\delta$$

due to Assumption 3 and subadditivity. Thus, we have that

$$\|\nabla_\theta l(\pi_{\theta_i}, \psi)\| \leq 2G\delta.$$

The assumptions in this section and the loss formulation are consistent with those in Hazan et al. (2016) and Osa et al. (2018) for analysis of online projected gradient descent and imitation learning algorithms.

3.3. Regret Analysis

To motivate FIT and the idea of learning from an improving supervisor, we show that given that the learner is represented by a strongly convex policy and satisfies other standard conditions, OGD has sublinear static regret with respect to the best policy in hindsight with labels from the final supervisor policy. Hazan et al. (2016) derive sublinear regret guarantees for OGD under similar assumptions with a static supervisor; we extend this analysis by showing that the additional asymptotic regret depends only on the convergence rate of the supervisor. Note that this is different from the type of regret analysis in (Ross et al., 2011a) and other on-policy imitation learning algorithms since in this setting, labels come not from a fixed supervisor, but from an improving supervisor during learning.

Theorem 1. *For loss function $l_i(\pi_{\theta_i}, \psi)$ satisfying the above assumptions, the expected static regret of OGD with labels from $(\psi_i)_{i=1}^N$ can be bounded above as follows:*

$$Regret_N \leq \frac{2G^2\delta^2}{\alpha}\Big(1 + \log N\Big) \\ + 2GD\sum_{i=1}^{N} \mathbb{E}_{s \in \{X_i\}} \|\psi_N(s) - \psi_i(s)\|$$

Proof of Theorem 1 can be found in the appendix.

Theorem 2. *If $\mathbb{E}_{s \in \{X_i\}} \|\psi_i(s) - \psi_N(s)\| \leq f_i$ w.p. 1 for any sequence $(f_i)_{i=1}^N$ where $\lim_{i\to\infty} f_i = 0$, the expected static regret of OGD is asymptotically sublinear:*

$$\sum_{i=1}^{N} \mathbb{E}_{s \in \{X_i\}} \|(\psi_N(s) - \psi_i(s)\| = o(N)$$

Proof of Theorem 2 can be found in the appendix.

Given Theorem 1, if we choose $f_i = \frac{C}{i}$ for $C \in \mathbb{R}^+$, we pay no extra asymptotic penalty in regret, achieving a regret bound of

$$\left(\frac{2G^2\delta^2}{\alpha} + 2GDC\right)\Big(1 + \log N\Big) \qquad (1)$$

We also note that any faster rate than $f_i = \frac{C}{i}$ also avoids additional asymptotic regret.

This motivates further exploration of on-policy imitation learning algorithms in the improving supervisor setting. See appendix for experimental studies on empirical regret and the effect of label quantity and quality on learning results.

4. Follow the Improving Teacher (FIT)

Motivated by the analysis in Section 3, we present Follow the Improving Teacher (FIT), which enables accelerated policy evaluation for deep model-based reinforcement learning. Instead of collecting policy rollouts using an an algorithmic supervisor that is expensive to query online, FIT uses a model-free policy to collect rollouts on each iteration, but labels for each state in the rollout are provided by the supervisor to refit the policy. FIT can be thought of as a meta-algorithm, which updates the model-free policy and algorithmic supervisor at each iteration using updates given by an on-policy imitation learning algorithm trained via labels from an off-policy reinforcement learning algorithm, which itself is updated using data from the model-free rollouts. Note that if the algorithmic supervisor has slow policy evaluation, such a procedure can speed up policy learning since the supervisor can label each state in a given rollout in parallel after the rollout has completed rather than serially at each timestep of every policy rollout. If using deep MPC algorithms such as those presented in (Chua et al., 2018; Nagabandi et al., 2018) as the algorithmic supervisor, the supervisor is updated by updating the dynamics model used for MPC with transitions collected from the learner rollouts. In general, any off-policy algorithmic supervisor can be used. There are also many possible choices for the on-policy imitation learning algorithm, such as OGD or DAgger. This is illustrated in Algorithm 1 below.

Algorithm 1 Follow the Improving Teacher (FIT)

```
Require: Randomly initialized off-policy algorithmic supervisor, $\psi_0$, model-free policy $\pi_{\theta_0}$
for $i \in \{1, \dots, N\}$ do
    Sample $T$ step trajectory from $\pi_{\theta_i}$
    Get dataset $\mathcal{D}_i = \{(s, \psi_i(s))\}$
    Update $\pi_{\theta_{i+1}}$ and $\psi_{i+1}$ using $\mathcal{D}_i$
end for
return $\pi_{\theta_N}$
```

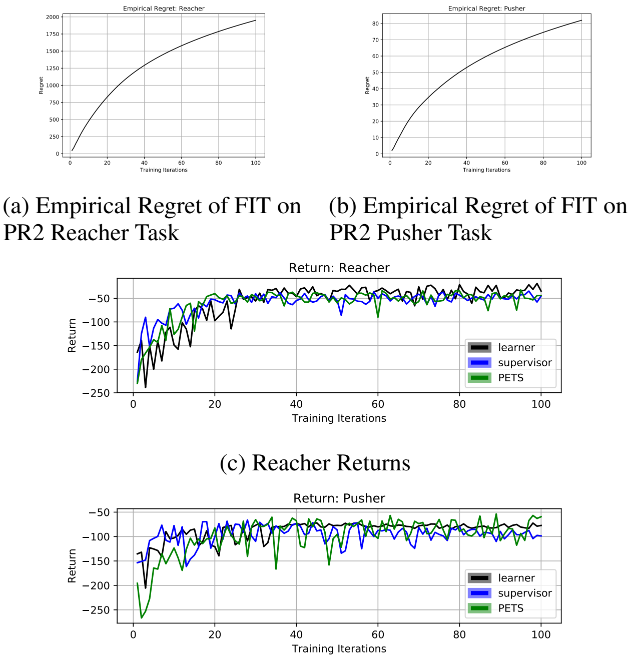
5. Experiments

In all experiments, we run FIT using the model-based deep reinforcement learning algorithm, PETS (Chua et al., 2018), as the improving supervisor, DAgger as the on-policy imitation learning algorithm, and squared loss between learner and supervisor as the imitation learning loss function. We consider the PR2 Reacher and Pusher MuJoCo domains from (Chua et al., 2018). See the appendix for further details on the parameters used for PETS.

For experiments, task return is reported for FIT, the supervisor, and PETS. Returns for FIT and the supervisor are computed by rolling out the model-free learner policy and model-based controller after each training iteration and computing the task return respectively. Note that the supervisor here is trained on off-policy data from the learner, so the difference between the learner and supervisor performance measures how effectively the learner is able to track the supervisor performance. Furthermore, we also report task return for the original PETS algorithm, which is trained on data from its own policy. Thus, the difference in performance between the supervisor and PETS measures how important on-policy data is for supervisor performance.

In Figure 1, we show the empirical regret and return for FIT, where the learner is represented with a linear policy trained via ridge-regression with regularization parameter $\alpha = 1$.

We observe that the empirical regret shows a sub-linear growth pattern for both Reacher and Pusher as shown in Figures 1a and 1b. We also see in Figures 1c and 1d that for both tasks, that not only is the learner able to successfully track the supervisor, but also that the supervisor performance is not significantly harmed by training on off-policy data in



(a) Empirical Regret of FIT on PR2 Reacher Task

(b) Empirical Regret of FIT on PR2 Pusher Task

(c) Reacher Returns

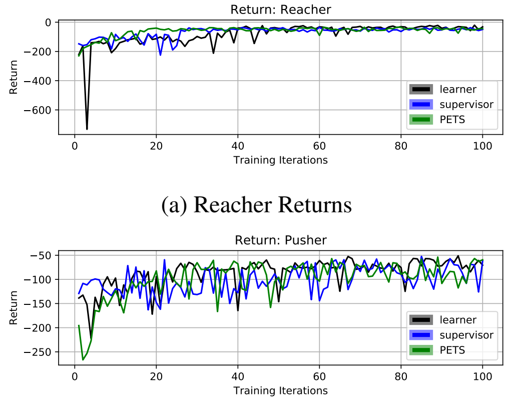
(d) Pusher Returns

Figure 1. **FIT with a linear learner policy:** We first show the empirical regret of FIT for (a) Reacher and (b) Pusher, which show clearly sublinear growth as expected. We also show training curves for the learner, supervisor and PETS on (c) Reacher and (d) Pusher. FIT is not only able to successfully track the supervisor on both domains, but also performs well compared to PETS. However, performance is slightly better on Reacher.

these settings. However, we notice that for the Pusher task, both the learner and supervisor converge to a slightly worse final return than PETS.

Finally, to determine whether FIT can effectively scale to more complex learner policy representations and to determine whether a more expressive learner could close the gap between FIT and PETS on the Pusher task, we repeat the above experiments with the learner represented by a neural network with 2 hidden layers with 20 hidden units each, ReLU activations, and trained using the Adam optimizer. Results are shown in Figure 2. The learner is able to match the supervisor and PETS closely for both tasks, demonstrating the efficacy of FIT in these settings.

This result is promising because if the model-free learner policy is able to achieve similar performance when tracking a model-based reinforcement learning supervisor compared to the supervisor on its own distribution, we achieve the sample complexity benefits of model-based reinforcement learning while achieving the low online computation time of model-free methods. This has potential to accelerate both training and testing times for model-based reinforcement learning algorithms by simply labeling states visited by a model-free policy in parallel after each rollout. We hope to explore this further in future work. As an initial check to determine the potential speedup that FIT could provide, we measure the average rollout time of the model-based controller PETS and the model-free learner policy over 50 rollouts for the Reacher task on a desktop running Ubuntu 16.04 with a 3.60 GHz Intel Core i7-6850K, 12 core CPU and an NVIDIA GeForce GTX 1080. On the Reacher task, we find that PETS has an average rollout time of 25.25 seconds while the model-free policy has an average rollout time of 0.324 seconds, demonstrating that a significant speedup is possible with a fully parallel implementation.



(a) Reacher Returns

(b) Pusher Returns

Figure 2. **FIT with a neural network learner policy:** Training curves for the learner, supervisor and PETS on (a) Reacher and (b) Pusher. We see that FIT is not only able to successfully track the supervisor on both domains, but also performs well compared to PETS for both tasks.

6. Conclusion

We introduce a new setting for on-policy imitation learning in which the expert policy is not fixed, but improving over time. We show that if the learner policy is strongly convex, and has bounded parameter space diameter, action space, and Jacobian operator norm, OGD with labels from an improving supervisor yields sublinear regret with respect to the best policy in hindsight trained with labels from the final supervisor. We use this to motivate a new algorithm, FIT, which provides the sample complexity benefits of deep MBRL while enabling the fast policy evaluation time of model-free methods. In future work, we hope to provide regret analysis for stochastic supervisors with specific noise profiles and consider a more general class of surrogate loss functions. We are also interested in implementing FIT in a parallelized manner to show that learning from an improving supervisor allows substantial speedup in wall-clock time without significantly affecting task performance.

References

Bagnell, J. Andrew (Drew). An invitation to imitation. Technical Report CMU-RI-TR-15-08, Carnegie Mellon University, Pittsburgh, PA, March 2015.

Cheng, Ching-An and Boots, Byron. Convergence of value aggregation for imitation learning. *International Conference on Artificial Intelligence and Statistics*, abs/1801.07292, 2018. URL http://arxiv.org/abs/1801.07292.

Chua, Kurtland, Calandra, Roberto, McAllister, Rowan, and Levine, Sergey. Deep reinforcement learning in a handful of trials using probabilistic dynamics models. *NeurIPS*, abs/1805.12114, 2018. URL http://arxiv.org/abs/1805.12114.

Deisenroth, MP. and Rasmussen, CE. Pilco: A model-based and data-efficient approach to policy search. In *Proceedings of the 28th International Conference on Machine Learning, ICML 2011*, pp. 465–472. Omnipress, 2011.

Hazan, Elad et al. Introduction to online convex optimization. *Foundations and Trends® in Optimization*, 2(3-4): 157–325, 2016.

Kahn, Gregory, Zhang, Tianhao, Levine, Sergey, and Abbeel, Pieter. Plato: Policy learning using adaptive trajectory optimization. *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3342–3349, 2017.

Kakade, Sham M and Tewari, Ambuj. On the generalization ability of online strongly convex programming algorithms. In Koller, D., Schuurmans, D., Bengio, Y., and Bottou, L. (eds.), *Advances in Neural Information Processing Systems 21*, pp. 801–808. Curran Associates, Inc., 2009.

Khetan, Ashish, Lipton, Zachary C., and Anandkumar, Anima. Learning from noisy singly-labeled data. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=H1sUHgb0Z.

Laskey, Michael, Lee, Jonathan, Chuck, Caleb, Gealy, David, Hsieh, Wesley, Pokorny, Florian T, Dragan, Anca D, and Goldberg, Ken. Robot grasping in clutter: Using a hierarchy of supervisors for learning from demonstrations. *Automation Science and Engineering (CASE), 2016 IEEE*, pp. 827–834, 2016a.

Laskey, Michael, Staszak, Sam, Hsieh, Wesley Yu-Shu, Mahler, Jeffrey, Pokorny, Florian T, Dragan, Anca D, and Goldberg, Ken. Shiv: Reducing supervisor burden in dagger using support vectors for efficient learning from demonstrations in high dimensional state spaces. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 462–469. IEEE, 2016b.

Laskey, Michael, Chuck, Caleb, Lee, Jonathan, Mahler, Jeffrey, Krishnan, Sanjay, Jamieson, Kevin, Dragan, Anca, and Goldberg, Ken. Comparing human-centric and robot-centric sampling for robot deep learning from demonstrations. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 358–365. IEEE, 2017.

Le, Hoang M, Kang, Andrew, Yue, Yisong, and Carr, Peter. Smooth imitation learning for online sequence prediction. *arXiv preprint arXiv:1606.00968*, 2016.

Lee, Jonathan, Laskey, Michael, Tanwani, Ajay Kumar, Aswani, Anil, and Goldberg, Kenneth Y. A dynamic regret analysis and adaptive regularization algorithm for on-policy robot imitation learning. *WAFR*, 2019.

Levine, Sergey, Wagener, Nolan, and Abbeel, Pieter. Learning contact-rich manipulation skills with guided policy search. *CoRR*, abs/1501.05611, 2015. URL http://arxiv.org/abs/1501.05611.

Menda, Kunal, Driggs-Campbell, Katherine, and Kochenderfer, Mykel J. Dropoutdagger: A bayesian approach to safe imitation learning. *arXiv preprint arXiv:1709.06166*, 2017.

Nagabandi, Anusha, Kahn, Gregory, Fearing, Ronald S., and Levine, Sergey. Neural network dynamics for model-based deep reinforcement learning with model-free fine-tuning. *ICRA*, 2018.

Natarajan, Nagarajan, Dhillon, Inderjit S, Ravikumar, Pradeep K, and Tewari, Ambuj. Learning with noisy labels. In Burges, C. J. C., Bottou, L., Welling, M., Ghahramani, Z., and Weinberger, K. Q. (eds.), *Advances in Neural Information Processing Systems 26*, pp. 1196–1204. Curran Associates, Inc., 2013. URL http://papers.nips.cc/paper/5073-learning-with-noisy-labels.pdf.

Osa, Takayuki, Pajarinen, Joni, Neumann, Gerhard, Bagnell, J. Andrew, Abbeel, Pieter, and Peters, Jan. An algorithmic perspective on imitation learning. *Foundations and Trends in Robotics*, 2018. URL http://arxiv.org/abs/1811.06711.

Pan, Yunpeng, Cheng, Ching-An, Saigol, Kamil, Lee, Keuntak, Yan, Xinyan, Theodorou, Evangelos, and Boots, Byron. Agile autonomous driving via end-to-end deep imitation learning. In *Proceedings of Robotics: Science and Systems (RSS)*, 2018.

Pomerleau, Dean A. Alvinn: An autonomous land vehicle in a neural network. In *Advances in neural information processing systems*, pp. 305–313, 1989.

Raykar, Vikas C., Yu, Shipeng, Zhao, Linda H., Jerebko, Anna, Florin, Charles, Valadez, Gerardo Hermosillo, Bogoni, Luca, and Moy, Linda. Supervised learning from multiple experts: Whom to trust when everyone lies a bit. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, pp. 889–896, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-516-1. doi: 10.1145/1553374.1553488. URL http://doi.acm.org/10.1145/1553374.1553488.

Richardson, Matthew and Domingos, Pedro. learning-with-knowledge-from-multiple-experts. In *Proceedings of the Twentieth International Conference on Machine Learning*, pp. 624–631. AAAI Press, January 2003. URL https://www.microsoft.com/en-us/research/publication/learning-with-knowledge-from\-multiple-experts/.

Ross, Stéphane, Gordon, Geoffrey J., and Bagnell, J. Andrew. A reduction of imitation learning and structured prediction to no-regret online learning. In *AISTATS*, 2011a.

Ross, Stéphane, Gordon, Geoffrey J, and Bagnell, J Andrew. A reduction of imitation learning and structured prediction to no-regret online learning. *International Conference on Artificial Intelligence and Statistics*, 2011b.

Sun, Wen, Venkatraman, Arun, Gordon, Geoffrey J., Boots, Byron, and Bagnell, J. Andrew. Deeply AggreVaTeD: Differentiable imitation learning for sequential prediction. In Precup, Doina and Teh, Yee Whye (eds.), *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 3309–3318, International Convention Centre, Sydney, Australia, 06–11 Aug 2017. PMLR.

Yan, Yan, Rosales, Romer, Fung, Glenn, Farooq, Faisal, Rao, Bharat, and Dy, Jennifer. Active learning from multiple knowledge sources. In Lawrence, Neil D. and Girolami, Mark (eds.), *Proceedings of the Fifteenth International Conference on Artificial Intelligence and Statistics*, volume 22 of *Proceedings of Machine Learning Research*, pp. 1350–1357, La Palma, Canary Islands, 21–23 Apr 2012. PMLR. URL http://proceedings.mlr.press/v22/yan12.html.

Zhang, Chicheng and Chaudhuri, Kamalika. Active learning from weak and strong labelers. In Cortes, C., Lawrence, N. D., Lee, D. D., Sugiyama, M., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 28*, pp. 703–711. Curran Associates, Inc., 2015. URL http://papers.nips.cc/paper/5988-active-learning-from-weak-and-\strong-labelers.pdf.

Zinkevich, Martin. Online convex programming and generalized infinitesimal gradient ascent. In *Proceedings of the 20th International Conference on Machine Learning (ICML-03)*, pp. 928–936, 2003.

7. Appendix

7.1. Proof of Theorem 1

We proceed similarly to Hazan et al. (2016). The parameter update via online projected gradient descent is given by:

$$\begin{aligned}&\|\theta_{i+1}-\theta^\star\|^2 = \\ &\|\Pi_{\mathcal{K}}(\theta_i - \eta_i \nabla_\theta l_i\left(\pi_{\theta_i}, \psi_i\right)) - \theta^\star\|^2 \leq \\ &\|\theta_i - \eta_i \nabla_\theta l_i\left(\pi_{\theta_i}, \psi_i\right) - \theta^\star\|^2\end{aligned} \tag{2}$$

Expanding the above gives:

$$\begin{aligned}&\|\theta_{i+1}-\theta^\star\|^2 \leq \\ &\|\theta_i - \theta^\star\|^2 + \eta_i^2 \|\nabla_\theta l_i\left(\pi_{\theta_i}, \psi_i\right)\|^2 - \\ &2\eta_i (\nabla_\theta l_i\left(\pi_{\theta_i}, \psi_i\right))^T (\theta_i - \theta^\star)\end{aligned} \tag{3}$$

Rearranging this gives:

$$\begin{aligned}&2(\nabla_\theta l_i\left(\pi_{\theta_i}, \psi_i\right))^T(\theta_i - \theta^\star) \leq \\ &\frac{\|\theta_i - \theta^\star\|^2 - \|\theta_{i+1} - \theta^\star\|^2}{\eta_i} + 4\eta_i G^2 \delta^2\end{aligned} \tag{4}$$

where $\|\nabla_\theta l_i\left(\pi_{\theta_i}, \psi_i\right)\| \leq 2G\delta$ by Assumption 4.

Given that the loss is of form:

$$l_i(\pi_{\theta_i}, \psi_i) = \mathbb{E}_{s \in \{X_i\}} \|\pi_{\theta_i}(s) - \psi_i(s)\|^2$$

we can relate the gradients of the loss with labels from the final supervisor ψ_N and from the labeler at iteration i (ψ_i) as follows:

$$\begin{aligned}&\nabla_\theta l_i\left(\pi_{\theta_i}, \psi_i\right) = \\ &\mathbb{E}_{s \in \{X_i\}}\left[2(\nabla_\theta \pi_{\theta_i}(s))^T\left(\pi_{\theta_i}(s) - \psi_i(s)\right)\right]\end{aligned} \tag{5}$$

$$\begin{aligned}&\nabla_\theta l_i\left(\pi_{\theta_i}, \psi_N\right) = \\ &\mathbb{E}_{s \in \{X_i\}}\left[2(\nabla_\theta \pi_{\theta_i}(s))^T\left(\pi_{\theta_i}(s) - \psi_N(s)\right)\right]\end{aligned} \tag{6}$$

We can compute the difference of the loss gradients as follows:

$$\begin{aligned}&\nabla_\theta l_i\left(\pi_{\theta_i}, \psi_i\right) - \nabla_\theta l_i\left(\pi_{\theta_i}, \psi_N\right) = \\ &\mathbb{E}_{s \in \{X_i\}}\left[2(\nabla_\theta \pi_{\theta_i}(s))^T\left(\psi_N(s) - \psi_i(s)\right)\right]\end{aligned} \tag{7}$$

so

$$\begin{aligned}&\nabla_\theta l_i\left(\pi_{\theta_i}, \psi_i\right) = \nabla_\theta l_i\left(\pi_{\theta_i}, \psi_N\right) + \\ &\mathbb{E}_{s \in \{X_i\}}\left[2(\nabla_\theta \pi_{\theta_i}(s))^T\left(\psi_N(s) - \psi_i(s)\right)\right]\end{aligned} \tag{8}$$

Thus, by substituting the right-hand side of equation 8 into the left-hand side of equation 4, we obtain:

$$\begin{aligned}&2\Big(\nabla_\theta l_i\left(\pi_{\theta_i}, \psi_N\right) \\ &+ \mathbb{E}_{s \in \{X_i\}}\left[2(\nabla_\theta \pi_{\theta_i}(s))^T\left(\psi_N(s) - \psi_i(s)\right)\right]\Big)^T(\theta_i - \theta^\star) \leq \\ &\frac{\|\theta_i - \theta^\star\|^2 - \|\theta_{i+1} - \theta^\star\|^2}{\eta_i} + 4\eta_i G^2 \delta^2\end{aligned} \tag{9}$$

We can rearrange the above to:

$$\begin{aligned}&2(\nabla_\theta l_i\left(\pi_{\theta_i}, \psi_N\right))^T(\theta_i - \theta^\star) \leq \\ &\frac{\|\theta_i - \theta^\star\|^2 - \|\theta_{i+1} - \theta^\star\|^2}{\eta_i} + 4\eta_i G^2 \delta^2 + \\ &\mathbb{E}_{s \in \{X_i\}}\left[4(\nabla_\theta \pi_{\theta_i}(s))^T\left(\psi_N(s) - \psi_i(s)\right)\right]^T(\theta^\star - \theta_i)\end{aligned} \tag{10}$$

Applying Cauchy–Schwarz and subadditivity gives:

$$\begin{aligned}&\mathbb{E}_{s \in \{X_i\}}\left[4(\nabla_\theta \pi_{\theta_i}(s))^T\left(\psi_N(s) - \psi_i(s)\right)\right]^T(\theta^\star - \theta_i) \leq \\ &\qquad \mathbb{E}_{s \in \{X_i\}} 4\|\nabla_\theta \pi_{\theta_i}(s))\|\|\psi_N(s) - \psi_i(s)\|\|\theta^\star - \theta_i\|\end{aligned} \tag{11}$$

where $\|\nabla_\theta \pi_{\theta_i}(s))\|$ is the operator norm of $\nabla_\theta \pi_{\theta_i}(s)$.

Using Assumption 2 and Assumption 4, we have that

$$\begin{aligned}&\mathbb{E}_{s \in \{X_i\}} 4\|\nabla_\theta \pi_{\theta_i}(s)\|\|\psi_N(s) - \psi_i(s)\|\|\theta^\star - \theta_i\| \leq \\ &4D\mathbb{E}_{s \in \{X_i\}}\|\nabla_\theta \pi_{\theta_i}(s))\|\|\psi_N(s) - \psi_i(s)\| \leq \\ &4GD\mathbb{E}_{s \in \{X_i\}}\|\psi_N(s) - \psi_i(s)\|\end{aligned} \tag{12}$$

Thus, we can rewrite equation 10 as follows:

$$\begin{aligned}&\nabla_\theta l_i\left(\pi_{\theta_i}, \psi_N\right)^T\left(\theta_i - \theta^\star\right) \leq \\ &\frac{\|\theta_i - \theta^\star\|^2 - \|\theta_{i+1} - \theta^\star\|^2}{2\eta_i} + 2\eta_i G^2 \delta^2 \\ &+ 2GD\mathbb{E}_{s \in \{X_i\}}\|(\psi_N(s) - \psi_i(s))\|\end{aligned} \tag{13}$$

Now, we can use the strong convexity of the loss function (Assumption 1) to obtain the following bound for $\alpha > 0$:

$$\begin{aligned}&\nabla_\theta l_i\left(\pi_{\theta_i}, \psi_N\right)^T\left(\theta_i - \theta^\star\right) - \frac{\alpha}{2}\|\theta_i - \theta^*\|_2^2 \geq \\ &l_i\left(\pi_{\theta_i}, \psi_N\right) - l_i\left(\pi_{\theta^\star}, \psi_N\right)\end{aligned} \tag{14}$$

Now using strong convexity and equation 13 gives:

$$\begin{aligned}&l_i\left(\pi_{\theta_i}, \psi_N\right) - l_i\left(\pi_{\theta^\star}, \psi_N\right) \leq \\ &\frac{\|\theta_i - \theta^\star\|^2 - \|\theta_{i+1} - \theta^\star\|^2}{2\eta_i} + 2\eta_i G^2 \delta^2 + \\ &2GD\mathbb{E}_{s \in \{X_i\}}\|\psi_N(s) - \psi_i(s)\| - \frac{\alpha}{2}\|\theta_i - \theta^*\|_2^2\end{aligned} \tag{15}$$

Now, summing each side over the iterations gives:

$$\begin{aligned}&\sum_{i=1}^{N}\left[l_i\left(\pi_{\theta_i}, \psi_N\right) - l_i\left(\pi_{\theta^\star}, \psi_N\right)\right] \leq \\ &\sum_{i=1}^{N}\Bigg[\frac{\|\theta_i - \theta^\star\|^2 - \|\theta_{i+1} - \theta^\star\|^2}{2\eta_i} + 2\eta_i G^2 \delta^2 \\ &+ 2GD\mathbb{E}_{s \in \{X_i\}}\|\psi_N(s) - \psi_i(s)\| - \frac{\alpha}{2}\|\theta_i - \theta^*\|_2^2\Bigg]\end{aligned} \tag{16}$$

Now, we can explicitly represent this in terms of a telescoping sum as follows:

$$\begin{aligned}&\sum_{i=1}^{N}\Bigg[\frac{\|\theta_i-\theta^\star\|^2-\|\theta_{i+1}-\theta^\star\|^2}{2\eta_i}+2\eta_iG^2\delta^2\\&+2GD\mathbb{E}_{s\in\{X_i\}}\|\psi_N(s)-\psi_i(s)\|-\frac{\alpha}{2}\|\theta_i-\theta^*\|_2^2\Bigg]\le\\&\sum_{i=1}^{N}\Bigg[\frac{\|\theta_i-\theta^\star\|^2}{2}\left(\frac{1}{\eta_i}-\frac{1}{\eta_{i-1}}-\alpha\right)+2\eta_iG^2\delta^2\\&+2GD\mathbb{E}_{s\in\{X_i\}}\|\psi_N(s)-\psi_i(s)\|\Bigg]\end{aligned}\tag{17}$$

Using $\eta_i=\frac{1}{\alpha i}$ for $i>0$ and $\frac{1}{\eta_0}=0$, we see that

$$\sum_{i=1}^{N}\left[\frac{\|\theta_i-\theta^\star\|^2}{2}\left(\frac{1}{\eta_i}-\frac{1}{\eta_{i-1}}-\alpha\right)\right]=0$$

Thus, we have:

$$\begin{aligned}&\sum_{i=1}^{N}\left[l_i\left(\pi_{\theta_i},\psi_N\right)-l_i\left(\pi_{\theta^\star},\psi_N\right)\right]\le\\&2G^2\delta^2\sum_{i=1}^{N}\eta_i+2GD\sum_{i=1}^{N}\mathbb{E}_{s\in\{X_i\}}\|\psi_N(s)-\psi_i(s)\|\end{aligned}\tag{18}$$

so

$$\begin{aligned}&\mathrm{Regret_N}=\sum_{i=1}^{N}\left[l_i\left(\pi_{\theta_i},\psi_N\right)-l_i\left(\pi_{\theta^\star},\psi_N\right)\right]\le\\&\frac{2G^2\delta^2}{\alpha}\Big(1+\log N\Big)\\&+2GD\sum_{i=1}^{N}\mathbb{E}_{s\in\{X_i\}}\|\psi_N(s)-\psi_i(s)\|\quad\square\end{aligned}\tag{19}$$

7.2. Proof of Theorem 2

If

$$\mathbb{E}_{s\in\{X_i\}}\|\psi_i(s)-\psi_N(s)\|\le f_i$$

w.p. 1 for any sequence $(f_i)_{i=1}^N$ where $\lim_{i\to\infty}f_i=0$, then this implies that

$$\sum_{i=1}^{N}\mathbb{E}_{s\in\{X_i\}}\|(\psi_N(s)-\psi_i(s)\|\le\sum_{i=1}^{N}f_i$$

The Additive Cesàro's Theorem states that if the sequence $(a_n)_n^\infty$ has a limit, then

$$lim_{n\to\infty}\frac{a_1+a_2\ldots a_n}{n}=lim_{n\to\infty}a_n$$

Thus, we see that if $\lim_{i\to\infty}f_i=0$, then it must be the case that $\lim_{N\to\infty}\frac{1}{N}\sum_{i=1}^{N}f_i=0$. This shows that for any $(f_i)_{i=1}^N$ converging to 0, it must be the case that

$$\sum_{i=1}^{N}\mathbb{E}_{s\in\{X_i\}}\|\psi_N(s)-\psi_i(s)\|\le\sum_{i=1}^{N}f_i=\mathcal{O}(N)$$

Thus, based on the regret bound in Theorem 1, we can achieve sublinear regret for any sequence $(f_i)_{i=1}^N$ which converges to 0. □

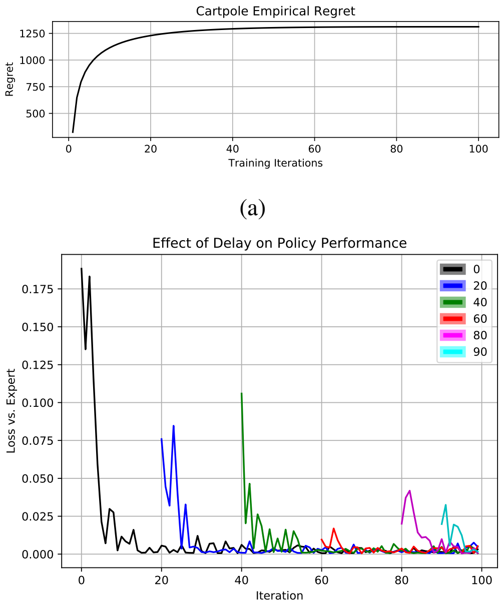
7.3. PETS Details

PETS learns an ensemble of neural network dynamics models using sampled transitions and updates them on-policy to better reflect the dynamics local to the learned policy's state distribution. MPC is run over the learned dynamics to select actions for the next iteration. For both the Reacher and Pusher environments, an ensemble of 5 neural networks with 3 hidden layers, each with 500 hidden units are used to represent the dynamics model. We use an MPC planning horizon of length 25 for both environments. Chua et al. (2018) contains further details on training PETS.

7.4. Experimental Study of OGD on Cartpole with iLQR Expert

Here, we study the performance of FIT on a Cartpole task with known dynamics. We use a linear policy trained via ridge-regression with regularization parameter $\alpha=1$ to represent the learner and an iLQR controller for the supervisor. Since the iLQR controller has good global performance in this domain, we study the effect of a specific supervisor convergence rate by explicitly adding progressively smaller action bias to the iLQR controller to simulate a supervisor improving at a specific rate. OGD is used for policy updates, so this experiment exactly satisfies the assumptions of the theoretical analysis. For these experiments, we simulate a supervisor improvement rate corresponding to $f_i=\frac{C}{\sqrt{i}}$, which gives expected regret on the order of $\mathcal{O}(\sqrt{N})$.

We show the empirical regret with an improving iLQR controller on Cartpole in Figure 3a. As expected, the regret shows a clear sublinear pattern. Furthermore, to study the effect of poor quality initial labels on learner performance, we show the average squared difference between the learner and supervisor actions for a set of learners that are each initialized at progressively later OGD iterations. The idea here is that learners which are initialized later will have access to less data, but this data will be from a higher quality supervisor. Thus, we expect to see a tradeoff between data quality and quantity, where sufficient amounts of low quality data may actually mislead the learner even if there is enough total data, while a very small amount of high quality data may be insufficient to successfully match supervisor performance.



(a)

(b)

Figure 3. **Performance of OGD on Cartpole with known dynamics:** (a) Empirical Regret of OGD on Cartpole with known dynamics and an iLQR teacher with action bias chosen to match supervisor convergence rate given by $f_i = \frac{C}{\sqrt{i}}$. Here, we see that the regret shows a clear sublinear pattern as expected; (b) Average squared difference between actions chosen by the learner and supervisor for a set of learners that are each initialized at progressively later OGD iterations. As expected, learners that are initialized earlier on take longer to converge and exhibit more noisy performance due to the low quality initial supervisor labels, while learners that are initialized later converge more quickly.

Results are shown in Figure 3b. Learners that are initialized earlier take longer to converge and still exhibit relatively noisy performance while those initialized later on appear to converge relatively quickly despite having fewer supervisor labels. It is possible that on a more difficult domain, fewer supervisor labels would have a more adverse effect on learner performance. We hope to investigate this tradeoff between data quality and quantity, specifically with regards to how a learner can best determine which supervisor labels to use when the supervisor is time-varying, in future work.

7.5. FIT Experimental Results on Reacher/Pusher

The final return achieved by FIT, the supervisor, and PETS corresponding to Figures 1 and 2 is shown in Table 1. Since FITis able to match the performance PETS relatively closely even when trained with completely off-policy data, this indicates that there is significant potential for accelerating learning and policy evaluation without significantly affecting task performance.

Table 1. **Learner, Supervisor, and PETS final training rewards:** We report (learner final reward, supervisor final reward, PETS final reward) for the Reacher and Pusher tasks after 100 training iterations with a linear policy and neural network policy. Although there is no significant difference in the policy representations for Reacher, for Pusher the increased expressiveness of the neural network policy does improve the learner's performance. For both environments, the learner is able to achieve a final return within 16 % of PETS, even with completely off-policy data.

Task	Reacher	Pusher
Linear Policy	(-34.70, -45.20, -44.08)	(-77.33, -98.63, -59.41)
Neural Net Policy	(-31.67, -49.21, -44.08)	(-70.48, -61.72, -59.41)