A Dynamic Regret Analysis and Adaptive Regularization Algorithm for On-Policy Robot Imitation Learning

Jonathan Lee, Michael Laskey, Ajay Kumar Tanwani,
Anil Aswani, and Ken Goldberg

University of California, Berkeley
{jonathan_lee, laskeymd, ajay.tanwani, aaswani, goldberg}@berkeley.edu

Abstract. On-policy imitation learning algorithms such as DAGGER evolve a robot control policy by executing it, measuring performance (loss), obtaining corrective feedback from a supervisor, and generating the next policy. As the loss between iterations can vary unpredictably, a fundamental question is under what conditions this process will eventually achieve a converged policy. If one assumes the underlying trajectory distribution is static (stationary), it is possible to prove convergence for DAGGER. Cheng and Boots (2018) consider the more realistic model for robotics where the underlying trajectory distribution, which is a function of the policy, is dynamic and show that it is possible to prove convergence when a condition on the rate of change of the trajectory distributions is satisfied. In this paper, we reframe that result using dynamic regret theory from the field of Online Optimization to prove convergence to locally optimal policies for DAGGER, Imitation Gradient, and Multiple Imitation Gradient. These results inspire a new algorithm, Adaptive On-Policy Regularization (AOR), that ensures the conditions for convergence. We present simulation results with cart-pole balancing and walker locomotion benchmarks that suggest AOR can significantly decrease dynamic regret and chattering. To our knowledge, this the first application of dynamic regret theory to imitation learning.

Keywords: Machine Learning, Optimization, Imitation Learning

1 Introduction

In imitation learning, the robot observes states and control labels from a supervisor and estimates a mapping from states to controls. A fundamental problem in imitation learning is covariate shift [1], where the distribution of trajectories experienced by the robot at run time differs from the distributions experienced during training time. For example, consider an autonomous vehicle trained to drive on a road from demonstrations of humans driving safely on the center of the same road. If the vehicle makes slight errors when it is deployed, it may drift towards the sides of the road where it had not previously experienced data from

human supervisors, resulting in poor performance leading to drift from which it cannot recover.

On-policy Imitation learning algorithms such as DAGGER [13], AGGREVATED [14], and LOKI [4] have been proposed to mitigate this issue. As opposed to learning only from supervisor demonstrations, these algorithms roll out the robot's current policy at each iteration, allowing it to make errors and observe new states. The supervisor then provides corrective control labels for these new states retroactively. For this reason, these algorithms are often referred to as on-policy imitation learning algorithms because the robot iteratively learns from its current policy [11]. This is in contrast to off-policy algorithms which learn from the supervisor's demonstrations. Recently, there has been interest in determining when on-policy algorithms are guaranteed to converge to good policies because practitioners often settle on the policy from the final iteration of the algorithm. Cheng and Boots [2] proved that DAGGER converges under the condition that a sensitivity parameter related to the rate of change of the trajectory distributions is small.

On-policy algorithms can be viewed as variants of algorithms from the field of Online Optimization [6]. Online Optimization is often used for problems such as portfolio management and network routing analysis where environments change over time [8]. At iteration n an agent plays some parameter θ_n and then some loss function f_n is presented. The agent then incurs loss $f_n(\theta_n)$. In on-policy imitation learning, θ_n would correspond to policy parameters. f_n would be a supervised learning loss function obtained from rolling out θ_n and observing corrective labels from the supervisor. At each iteration the supervised learning loss function is different because the distribution of trajectories induced by the robot changes as the policy is updated. The goal is to continually try to find the optimal θ_n at each iteration based on the sequence of past loss functions. A common choice in Online Optimization to measure the performance of an algorithm is static regret over N iterations, defined as

$$R_S(\theta_1, \ldots, \theta_N) := \sum_{n=1}^{N} f_n(\theta_n) - \min_{\theta} \sum_{n=1}^{N} f_n(\theta). \tag{1}$$

In imitation learning, this would mean the robot is compared against the best it could have done on the average of its trajectory distributions.

In this paper, we focus on determining when on-policy algorithms can and cannot converge and specifically when the policy is performing optimally on its own distribution. We draw a connection between this very natural objective and an alternative metric in Online Optimization known as *dynamic regret* [5,10,15]. As opposed to the well known static regret in (1), dynamic regret measures performance of a policy at each instantaneous iteration:

$$R_D(\theta_1, \ldots, \theta_N) := \sum_{n=1}^{N} f_n(\theta_n) - \sum_{n=1}^{N} \min_{\theta} f_n(\theta). \tag{2}$$

The difference between static and dynamic regret is that, for dynamic regret, the minimum goes inside the summation, meaning that the regret is an instanta-

neous difference at each iteration. Proving that static regret is low implies that the policy on average is at least as good as a single fixed policy that does well on the average of the distributions seen during training. Proving that dynamic regret is low implies that the average policy is as good as possible on its own distribution. That is, the policy is locally optimal. Achieving low dynamic regret is inherently harder than achieving low static regret because $R_S \leq R_D$, but dynamic regret is more relevant as a theoretical metric in robotics where the trajectory distributions are changing.

In Online Optimization, it is well known that it is not possible to achieve low dynamic regret in general due to the possibility of adversarial loss functions [10]. However, in imitation learning, the key insight is that the loss functions at each iteration represent the trajectory distributions as a function of the policy parameters [2]. Therefore, we can leverage information known about how the trajectory distribution changes in response to changing policy parameters to obtain specific dynamic regret rates.

This paper makes four contributions:

1. Introduces a dynamic regret analysis to evaluate the convergence of on-policy imitation learning algorithms.
2. Presents average dynamic regret rates and conditions for convergence for DAGGER, Imitation Gradient, and Multiple Imitation Gradient.
3. Introduces Adaptive On-Policy Regularization (AOR), a novel algorithm that adaptively regularizes on-policy algorithms to improve dynamic regret and cause convergence.
4. Presents empirical evidence of non-convergent on-policy algorithms and shows that AOR can ensure convergence in a cart-pole balancing task and a walker locomotion task.

2 Preliminaries

In this section we introduce the notation, problem statement, and assumptions for imitation learning by supervised learning. Let $s_t \in \mathcal{S}$ and $u_t \in \mathcal{U}$ be the state and control in a Markov decision process. The probability of a trajectory τ of length T under policy $\pi \in \Pi : \mathcal{S} \mapsto \mathcal{U}$ is given by

$$p(\tau; \pi) = p(s_0) \prod_{t=1}^{T-1} p_\pi(u_t|s_t) p(s_{t+1}|s_t, u_t).$$

In this paper, we consider parametric policies, i.e., there is a convex, bounded inner product space of parameters Θ with diameter $D := \max_{\theta_1, \theta_2 \in \Theta} \|\theta_1 - \theta_2\|$ and our goal is to find a parameterized policy π_θ that minimizes some loss with respect to the supervisor policy π^*, which may not necessarily be attainable by Θ. The loss of a policy π along a trajectory τ is a non-negative function J such that

$$J(\tau, \pi) = \sum_{t=1}^{T-1} \ell(\pi(s_t), \pi^*(s_t)),$$

where $\ell : \mathcal{U} \times \mathcal{U} \mapsto \mathbb{R}_{\geq 0}$ is a per-time step loss.

The general optimization problem of imitation learning can be written as

$$\min_{\theta \in \Theta} \mathbb{E}_{p(\tau;\pi_\theta)} J(\tau, \pi_\theta).$$

The expectation is taken over the distribution of trajectories that π_θ induces and then π_θ is evaluated on the trajectories sampled from that distribution. This problem reflects the goal of having the policy do well on its own induced distribution. However, this is challenging and cannot be solved with regular optimization methods since the distribution of trajectories is unknown. It also cannot be solved with regular supervised learning by sampling supervisor trajectories because the sampling distribution is a function of the policy [1].

Because this problem cannot be solved directly, existing algorithms relax it by fixing the trajectory distribution and then optimizing over the evaluation parameter. This decouples the sampling from the supervised learning problem. For example, in behavior cloning, one sets the trajectory distribution to the supervisor's trajectory distribution and finds the policy that minimizes loss on that distribution. Formally, we consider the average loss of a parameter $\theta \in \Theta$ over the distribution of trajectories generated by a possibly different policy parameter $\theta' \in \Theta$:

$$f_{\theta'}(\theta) := \mathbb{E}_{p(\tau;\pi_{\theta'})} J(\tau, \pi_\theta). \tag{3}$$

Here, θ' controls the trajectory distribution and θ controls the predictions used to the compute the loss on that distribution. For this reason, we refer to the θ' as the distribution-generating parameter and θ as the evaluation parameter. To better convey this notion of the policy parameters being decoupled, we use this concise f notation as in [2].

Optimization problems in this paper will be of the form $\min_{\theta \in \Theta} f_{\theta'}(\theta)$ for some fixed and known $\theta' \in \Theta$. Because the trajectory distribution no longer depends on the variable θ, the supervised learning problem can now be feasibly solved by sampling from $p(\tau; \pi_{\theta'})$ which corresponds to rolling out trajectories under the fixed policy $\pi_{\theta'}$. Specifically in this paper, we will consider iterative on-policy algorithms over $N \in \mathbb{N}$ iterations. At any iteration n for $1 \leq n \leq N$, the policy parameter θ_n is rolled out as the distribution-generating parameter and the loss $f_{\theta_n}(\theta) = \mathbb{E}_{p(\tau;\pi_{\theta_n})} J(\tau, \pi_\theta)$ is observed, where θ is the free variable and θ_n is the fixed per-iteration distribution-generating parameter. As in prior work [2,13], for convenience, we write $f_n(\theta) := f_{\theta_n}(\theta)$. These loss functions form the sequence of losses used in the regret metrics (1) and (2).

Next, we briefly describe the main assumptions of this paper. The assumptions are stated formally in Section 6. As in prior work in both imitation learning and Online Optimization, we assume strong convexity and smoothness of the loss function in the evaluation parameter. Strong convexity ensures the loss is curved at least quadratically while smoothness guarantees it is not too curved. As in [2], we also assume a regularity assumption on the f_n sequence, which bounds the sensitivity of the trajectory distribution in response to changes in the distribution-generating parameter.

3 Related Work

The challenge of covariate shift in imitation learning by supervised learning is the subject of significant research robotics. It is especially prevalent when the robot's policy cannot fully represent the supervisor [9]. In robotics, many algorithms have been proposed to mitigate covariate shift for imitation learning. Ross et al. [13] introduced DAGGER, an on-policy algorithm that allows the robot to make mistakes and then observe corrective labels from the supervisor in new states that might not be seen from ideal supervisor demonstrations alone.

Gradient-based on-policy methods for imitation learning have gained interest due to their similarity to policy gradient algorithms and their computational efficiency. These are also known as Imitation Gradient methods. AGGREVATED [14] was proposed for fast policy updates designed for deep neural network policies. LOKI [4] uses a mirror descent algorithm on an imitation learning loss to bootstrap reinforcement learning.

Ross et al. [13] first introduced a static regret-style analysis for DAGGER, showing that with strongly convex losses, running DAGGER results in low static regret in all cases. This means that a DAGGER policy is on average at least as good as one policy that does well on the average of trajectory distributions seen during training. However, the performance on the average of trajectory distributions is not always informative, as shown by Laskey et al. [9], because the average may contain irrelevant distributions as a result of rolling out suboptimal policies. Cheng and Boots [2] also recently expanded the DAGGER analysis showing that despite guaranteed convergence in static regret, the algorithm may not always converge to a low loss policy. Furthermore, they identified conditions sufficient for convergence to local optima. This work extends the results of Cheng and Boots [2] by drawing a connection with dynamic regret theory to identify conditions for convergence for DAGGER and other on-policy algorithms and as a basis for a new algorithm. To the best of our knowledge, this is the first application of dynamic regret theory to imitation learning.

4 On-Policy Algorithms

We now review three on-policy algorithms that will be the focus of the main theoretical and empirical results of the paper.

4.1 DAGGER

DAGGER is a variant of the follow-the-leader algorithm from Online Optimization. For detailed discussion of implementation, we refer the reader to [6,13]. DAGGER proceeds by rolling out the current policy and observing a loss based on the induced trajectory distribution. The next policy parameter is computed by aggregating all observed losses and minimizing over them. An example of this

Algorithm 1: DAGGER [13]

Input: Initial policy parameter θ_1,
Max iterations N.
for $n = 1$ **to** $N - 1$ **do**
Roll out θ_n and collect τ_n.
Form loss $f_n(\theta) = f_{\theta_n}(\theta)$ from supervisor feedback on τ_n.
$\theta_{n+1} \leftarrow \arg\min_{\theta \in \Theta} \sum_{m=1}^{n} f_m(\theta)$.
end for

Algorithm 2: (Multiple) Imitation Gradient [4,14,16]

Input: Initial policy parameter θ_1,
Max iterations N,
Updates per iteration K,
Stepsize η.
for $n = 1$ **to** $N - 1$ **do**
Roll out θ_n and collect τ_n.
Form loss $f_n(\theta) = f_{\theta_n}(\theta)$ from τ_n.
$\theta_n^1 \leftarrow \theta_n$
for $k = 1$ **to** K **do**
$\theta_n^{k+1} \leftarrow P_\Theta\left(\theta_n^k - \eta \nabla f_n(\theta_n^k)\right)$.
end for
$\theta_{n+1} \leftarrow \theta_n^{K+1}$.
end for

Left: DAGGER minimizes over all observed loss functions which are represented by a supervised learning loss over all observed data. Right: Multiple Imitation Gradient computes a gradient on only the most recent data collected and applies K gradient steps. Imitation Gradient is a special case where $K = 1$. P_Θ is a projection operation, projecting the resulting parameter vector back onto Θ in the event it lies outside.

for the l_2-regularized linear regression problem would be

$$\min_\theta \sum_{m=1}^{n} f_m(\theta) = \min_\theta \sum_{m=1}^{n} \mathbb{E}\|S_m\theta - U_m\|^2 + \frac{\alpha_0}{2}\|\theta\|^2, \tag{4}$$

where S_m is the matrix of state vectors observed from rolling out π_m and U_m is the matrix of labeled controls from the supervisor at the mth iteration.

4.2 Imitation Gradient and Multiple Imitation Gradient

Recently there has be interest in "Imitation Gradient" algorithms. Algorithms such as AGGREVATED and LOKI fall in this family. The online gradient descent algorithm from Online Optimization underlies such algorithms and their variants. While the aforementioned methods have explored more complicated gradient-based algorithms such as those inspired by the natural gradient and mirror descent, the analysis in this paper will focus on the basic online gradient descent algorithm. Online gradient descent proceeds by observing f_n at each iteration and taking a weighted gradient step: $\theta_n - \eta \nabla f_n(\theta_n)$. In the event that the resulting parameter lies outside of Θ, it is projected back on the space with the projection $P_\Theta(\theta) = \arg\min_{\theta' \in \Theta} \|\theta' - \theta\|$.

We also consider a very related algorithm termed Multiple Imitation Gradient, based on the online multiple gradient descent algorithm introduced by Zhang et al. [16]. This algorithm is identical to the Imitation Gradient, but at each iteration it updates the policy parameters K times, recomputing the gradient each time. Imitation Gradient is a special case of Multiple Imitation Gradient where $K = 1$. The algorithms are shown together in Algorithm 2.

5 Dynamic Regret

We are interested in showing that the policies generated by these algorithms perform well on the loss on their own induced trajectory distributions and, furthermore, that they converge. This means that we would like the difference $f_n(\theta_n) - \min_{\theta \in \Theta} f_n(\theta)$ to be as small as possible for every iteration n. This difference represents the instantaneous regret the algorithm has for playing θ_n instead of θ_n^* at iteration n where $\theta_n^* := \arg\min_{\theta \in \Theta} f_n(\theta)$. Summing over n from 1 to N, we have the definition of dynamic regret given in Equation (2).

The more well known static regret, shown in Equation (1), compares an algorithm's sequence of policies to the minimizer over the the average of losses on all trajectory distributions seen during training. Static regret has been shown to be low for strongly convex losses regardless of any external conditions such as the robot or system dynamics or distributions induced [13]. However, proving that a policy has low regret compared to the average of the trajectory distributions gives little indication of whether the policy actually performs well on its own induced distribution. The reason is that some distributions observed during training can be irrelevant to the task due to bad initialization or extremely sensitive dynamics, but they are still included in the average. As a result, static regret can be low regardless of actual policy performance or whether the algorithm leads to convergent policies. Prior work has shown in experiments and theoretical examples that on-policy algorithms do indeed fail on "hard" problems [2,9]. In order to characterize this notion of problem difficulty, we turn to dynamic regret.

In Online Optimization literature, dynamic regret has become increasingly popular to analyze online learning problems where the objective is constantly shifting [5,8,10,15,16,17]. For example in portfolio management, θ_n represents an investment strategy while f_n represents some negative return from the market. If the market is shifting over time, i.e. prices are changing, we are interested in playing the best strategy for the given state of the market at each time step. This can be represented as a dynamic regret problem. Dynamic regret compares the nth policy to the instantaneous best policy on the nth distribution, which means it is a stricter metric than static regret. The advantage of the dynamic regret metric that a policy's performance at any iteration is always evaluated with respect to the most relevant trajectory distribution, the current one. Proving dynamic regret is low implies that each policy on average is as good as the instantaneous best on its own distribution. Furthermore, we can examine convergence properties of an algorithm by proving that dynamic regret is low.

The dynamic regret of an algorithm is fundamentally dependent on the change in the loss functions over iterations, often expressed in terms of quantities called variations. If the loss functions change in an unpredictable manner, we can expect large variation terms leading to large regret and suboptimal policies. This is also the reason that sublinear dynamic regret bounds cannot be obtained in general using only the assumptions commonly used for static regret [15]. In this paper we consider the commonly used path variation and a squared variant of it introduced by [16].

Definition 1 (Path Variation and Squared Path Variation). *For a sequence of optimal parameters from m to n given by $\theta^*_{m:n} := (\theta^*_i)_{m \leq i \leq n}$, the path variation and the squared path variation are defined respectively as:*

$$V(\theta^*_{m:n}) := \sum_{i=m}^{n-1} \|\theta^*_i - \theta^*_{i+1}\| \quad \textit{and} \quad S(\theta^*_{m:n}) := \sum_{i=m}^{n-1} \|\theta^*_i - \theta^*_{i+1}\|^2.$$

Many algorithms have been proposed and analyzed in this new regret framework in terms of variation measures of the loss functions. For example, Zinkevich [17] proved a dynamic regret rate of $O(\sqrt{N}(1+V(\theta^*_{1:N})))$ for online gradient descent with convex losses. Here, the regret rate is dependent on the rate of the path variation, which might also be a function of N. Therein lies the difficulty of dynamic regret problems: no matter the algorithm, rates depend on the variation, which can be large for arbitrary sequences of loss functions.

Dynamic regret analyses offer a promising framework for theoretical analysis of on-policy imitation learning algorithms but only as long as the variation can be characterized. In imitation learning, the variation of the loss functions is related to the amount of change in the trajectory distributions induced by the sequence of policies. Ultimately these changes in trajectory distributions are dependent on the dynamics of the system. We will show in the next section that a single assumption on the dynamics, introduced by Cheng and Boots [2], can aid in characterizing the variation.

6 Main Results

In this section, we present the primary theoretical results of the paper, showing that convergence in average dynamic regret can be guaranteed for all three algorithms under certain conditions on the sensitivity of the trajectory distribution. We show that it is possible for all algorithms to achieve $O(N^{-1})$ average dynamic regret when these conditions are met. Since dynamic regret upper bounds static regret, these results suggest that we can also improve static regret rates of [13] by a logarithmic factor.

We first formally state the assumptions introduced in Section 2. We begin with common convex optimization assumptions on the loss in the evaluation parameter. Note that all gradients of f in this section are taken with respect to the evaluation parameter, i.e. $\nabla f_{\theta'}(\theta)$ denotes the gradient with respect to θ, not θ'.

Assumption 1 (Strong Convexity) *For all $\theta_1, \theta_2, \theta \in \Theta$, $\exists \alpha > 0$ such that*

$$f_\theta(\theta_2) \geq f_\theta(\theta_1) + \langle \nabla f_\theta(\theta_1), \theta_2 - \theta_1 \rangle + \frac{\alpha}{2}\|\theta_1 - \theta_2\|^2.$$

Assumption 2 (Smoothness and Bounded Gradient) *For all $\theta_1, \theta_2, \theta \in \Theta$, $\exists \gamma > 0$ such that*

$$\|\nabla f_\theta(\theta_1) - \nabla f_\theta(\theta_2)\| \leq \gamma \|\theta_1 - \theta_2\|$$

and $\exists G > 0$ such that $\|\nabla f_\theta(\theta_1)\| \leq G$.

Assumption 3 (Stationary Optimum) *For all $\theta' \in \Theta$, θ^* is in the relative interior of Θ where $\theta^* = \arg\min_{\theta \in \Theta} f_{\theta'}(\theta)$. That is, $\nabla f_{\theta'}(\theta^*) = 0$.*

In practice the above conditions are not difficult to satisfy. For example, running DAGGER with l_2-regularized linear regression, i.e. ridge regression, would simultaneously satisfy all three. Finally, we state the regularity constraint on the loss as a function of the distribution-generating parameter.

Assumption 4 (Regularity) *For all $\theta_1, \theta_2, \theta \in \Theta$, $\exists \beta > 0$ such that*

$$\|\nabla f_{\theta_1}(\theta) - \nabla f_{\theta_2}(\theta)\| \leq \beta \|\theta_1 - \theta_2\|.$$

This assumption is introduced as a form a prior knowledge of the dynamics. It is essentially a sensitivity constraint that implies we are assuming that small changes in the policy parameters guarantee small changes in the induced trajectory distributions. To prove low dynamic regret it is necessary that some prior knowledge be incorporated in order to simplify or regulate the variation measures. In this analysis, we use Assumption 4 because of its precedent in prior work [2].

We now present the main novel results of this paper for the infinite sample or deterministic case as in [2,13]. Let $\theta_n^* = \arg\min_{\theta \in \Theta} f_n(\theta)$ be the optimal parameter at iteration n. We begin with a result concerning a stability constant $\lambda := \frac{\beta}{\alpha}$ [2]. λ represents the ratio of the sensitivity and the strong convexity.

Proposition 1. *Given the assumptions, the following inequality holds on the difference between optimal parameters and corresponding policy parameters for any $n, m \in \mathbb{N}$:*

$$\|\theta_m^* - \theta_n^*\| \leq \lambda \|\theta_m - \theta_n\|.$$

The proof is in the appendix. Consider the case where $m = n + 1$. This proposition suggests that when $\lambda < 1$, we know with certainty that $\|\theta_{n+1}^* - \theta_n^*\| < \|\theta_{n+1} - \theta_n\|$. In other words, the optimal parameters cannot run away faster than the algorithm's parameters. This intuition is also consistent with the findings of prior work [2], which shows that convergence of the Nth policy can be guaranteed when $\lambda < 1$ for DAGGER.

6.1 DAGGER

We now introduce a dynamic regret corollary to Theorem 2 of [2].

Corollary 1. *For DAGGER under the assumptions, if $\lambda < 1$, then the average dynamic regret $\frac{1}{N} R_D$ tends towards zero in N with rate $O(\max(N^{-1}, N^{2\lambda - 2}))$.*

Proof. The proof is immediate from the result of Theorem 2 of [2]. We have $f_n(\theta_n) - f_n(\theta_n^*) \leq \frac{\left(\lambda e^{1-\lambda} G\right)^2}{2\alpha n^{2(1-\lambda)}}$. Summing from 1 to N, we get $\sum_{n=1}^{N} f_n(\theta_n) - \sum_{n=1}^{N} f_n(\theta_n^*) \leq \sum_{n=1}^{N} \frac{\left(\lambda e^{1-\lambda} G\right)^2}{2\alpha n^{2(1-\lambda)}} = O(\max(1, N^{2\lambda - 1}))$. Then the average dynamic regret is $\frac{1}{N} R_D = O(\max(N^{-1}, N^{2\lambda - 2}))$, which goes to zero. □

The corollary reveals that the convergence result for DAGGER proved by Cheng and Boots can be reframed as a dynamic regret analysis. The rate is dependent on the stability constant $\lambda < 1$. The dynamic regret shows that policies generated from DAGGER on average achieve local optimality and that for a sufficiently small λ, the regret grows no more than a finite amount.

6.2 Imitation Gradient

For the analysis of dynamic regret bounds for the Imitation Gradient algorithm, we require a stronger condition that $\alpha^2 > 2\gamma\beta$. Written another way, the condition is $2\lambda < \psi$ where λ is the stability constant and $\psi := \frac{\alpha}{\gamma}$ is the condition number of f_n. So we require that the problem is both stable and well-conditioned. In this proof, we will make use of the path variation.

Theorem 1. *For the Imitation Gradient algorithm under the assumptions, if* $\lambda < 1$, $2\lambda < \psi$ *and* $\eta = \frac{\alpha(\alpha^2 - 2\gamma\beta)}{2\gamma^2(\alpha^2 - \beta^2)}$, *then the average dynamic regret* $\frac{1}{N} R_D$ *tends towards zero in* N *with rate* $O(N^{-1})$. *Furthermore,* $\|\theta_n - \theta_n^*\|$ *converges to zero and the sequence of policies* $(\theta_n)_{n=1}^{\infty}$ *is convergent.*

The proof is in the appendix. Intuitively, the theorem states that if the conditions are met, the dynamic regret grows no more than a constant value. If we again interpret the variation as describing the difficulty of a problem, this theorem suggests that under the appropriate conditions, solving an imitation learning problem with Imitation Gradient algorithms is as easy as solving a general dynamic regret problem with path variation $V(\theta_{1:N}^*) = O(1)$. In other words, the equivalent dynamic regret problem is stationary in the limit: the optimal parameters cumulatively move no more than a finite distance. The reason is that the change in the loss functions is so closely tied to the policy parameters in imitation learning.

6.3 Multiple Imitation Gradient

We now present a similar dynamic regret rate for the Multiple Imitation Gradient algorithm. This theorem will make use of the squared path variation as opposed to the path variation. The squared path variation is especially amenable for a conversion from the standard dynamic regret bound given in [16] to an imitation learning analysis, as we will see in the proof of the theorem.

A straightforward conversion to characterize the measure of variation can be established by simply bounding from above the squared path variation by a quantity proportional to the dynamic regret using Assumption 4. We refer to this technique as establishing the reverse relationship between the measure of variation and the dynamic regret. To illustrate this conversion in detail, we present the proof here, which leverages the general dynamic regret rate for online multiple gradient descent first given by Zhang et al. [16]

Lemma 1 (Zhang et al. [16], Theorem 3). *If Assumptions 1-3 hold and* $\eta < 1/\gamma$ *and* $K = \lceil \frac{1/\eta+\alpha}{2\alpha} \log 4 \rceil$*, then the following is true for online multiple gradient descent:*

$$R_D(\theta_1, \dots \theta_N) \leq 2\gamma S(\theta^*_{1:N}) + \gamma \|\theta_1 - \theta^*_1\|^2.$$

From this lemma, a specific result for imitation learning can obtained in a straightforward manner by incorporating the regularity and the strong convexity of the loss functions.

Theorem 2. *For the Multiple Imitation Gradient algorithm under the assumptions, if* Θ *is Euclidean,* $\lambda < 1$*,* $\lambda \log 4 < \psi^{3/2}$*,* $\eta < \min\left\{1/\gamma, \frac{\alpha^{5/2} - \gamma^{3/2}\beta \log 4}{2\gamma^{3/2}\alpha\beta \log 4}\right\}$*, and* $K = \lceil \frac{1/\eta+\alpha}{2\alpha} \log 4 \rceil$ *then the average dynamic regret* $\frac{1}{N} R_D$ *tends towards zero in* N *with rate* $O(N^{-1})$*. Furthermore,* $\|\theta_n - \theta^*_n\|$ *converges to zero and the sequence of policies* $(\theta_n)_{n=1}^{\infty}$ *is convergent.*

Proof. In the interest of brevity, we skip some of the initial steps. The full proof can be in the appendix. We begin by establishing the reverse relationship; that is, we bound from above the squared path variation by a quantity proportional to the dynamic regret using Proposition 1 and Assumption 4:

$$S(\theta^*_{1:N}) \leq \frac{2\beta^2\eta^2\gamma^2K^2}{\alpha^3} \sum_{n=1}^{N} f_n(\theta_n) - f_n(\theta^*_n). \tag{5}$$

Then by substituting into Lemma 1, we have

$$\begin{aligned} R_D(\theta_1, \dots \theta_N) &\leq \gamma\|\theta_1 - \theta^*_1\|^2 + \frac{4\beta^2\eta^2\gamma^3K^2}{\alpha^3} \sum_{n=1}^{N} f_n(\theta_n) - f_n(\theta^*_n) \\ &\leq \frac{\gamma\|\theta_1 - \theta^*_1\|^2}{1 - \frac{4\beta^2\eta^2\gamma^3K^2}{\alpha^3}} \leq \frac{\gamma D^2}{1 - \frac{4\beta^2\eta^2\gamma^3K^2}{\alpha^3}}. \end{aligned}$$

It can be verified that for $\eta < \min\left\{1/\gamma, \frac{\alpha^{5/2} - \gamma^{3/2}\beta \log 4}{2\gamma^{3/2}\alpha\beta \log 4}\right\}$ and $K = \lceil \frac{1/\eta+\alpha}{2\alpha} \log 4 \rceil$, the denominator is positive. □

The above theorem demonstrates that again a constant upper bound on the dynamic regret can be obtained with this algorithm. Interestingly, the conditions sufficient to guarantee convergence are different. Instead of requiring $2\lambda < \psi$, this theorem requires $\lambda \log 4 < \psi^{3/2}$. While $\log 4 < 2$, we also have that $\psi^{3/2} \leq \psi$ because the condition number, which is the ratio of α and γ, is at most 1.

The implication of this observation is that there is a trade-off. For the Multiple Imitation Gradient algorithm, we can guarantee $O(N^{-1})$ regret for higher values of λ, i.e. more sensitive systems, but we must then require that our loss functions are better conditioned. Conversely, the Imitation Gradient algorithm achieves a similar guarantee using losses with lower condition numbers, but λ must be smaller.

7 Adaptive On-Policy Regularization

We now apply these theoretical results to motivate an adaptive regularization algorithm to help ensure convergence. As noted by Cheng et al. [2], regularization can lead to convergent policies for DAGGER.

Algorithm 3: Adaptive On-Policy Regularization (AOR)

Input: Initial parameters θ_1,
Max iterations N,
Initial regularization $\hat{\alpha}_1$.
for $n = 1$ **to** $N - 1$ **do**
Roll out θ_n and collect trajectory τ_n.
Observe loss $f_n(\theta) = f_{\theta_n}(\theta)$ from supervisor feedback on τ_n.
$\theta_{n+1} \leftarrow \textsc{Update}(\theta_n, \hat{\alpha}_n)$.
$\hat{\lambda}_n \leftarrow \textsc{Estimate}(f_1, \ldots, f_n)$.
$\hat{\alpha}_{n+1} \leftarrow \textsc{Tune}(\hat{\alpha}_n, \hat{\lambda}_n)$.
end for

Adaptive On-Policy Regularization adaptively increases α, the regularization parameter for strong convexity, to stabilize an on-policy algorithm. The algorithm can be used with any regularized on-policy algorithm by using the UPDATE subroutine.

In DAGGER, Imitation Gradient, and Multiple Imitation Gradient, a key sufficient condition is that $\lambda < 1$, meaning that the strong convexity constant α must be greater than the regularity constant β. While β is a fixed property of the dynamics, α is controllable by the user and robot. α can be increased by increasing the regularization of the supervised learning loss. By Proposition 1, a lower bound on λ may be estimated by finding the ratio of the distance between optimal parameters and the distance between policy parameters: $\hat{\lambda} = \frac{\|\theta^*_{n+1} - \theta^*_n\|}{\|\theta_{n+1} - \theta_n\|}$.

Therefore, we can propose an adaptive algorithm to compute a new regularization term at each iteration n. One caveat of adaptively updating α is that we do not want it to be too large. While the regret will converge, the policy performance can suffer as the regularization term will dominate the loss function and thus the convergent solution will simply be the solution that minimizes the regularizer. So we want α to be just large enough to converge but no larger. This subtlety and the the theoretical motivation in the previous subsections are the basis for Algorithm 3, which we call Adaptive On-Policy Regularization, an algorithm for making conservative updates to α that can be applied to any regularized on-policy algorithm. At each iteration, the policy is updated according to a given on-policy algorithm such as DAGGER using subroutine $\textsc{Update}(\theta_n, \hat{\alpha}_n)$ which depends on the current regularization. Then $\hat{\lambda}_n$ is computed with subroutine ESTIMATE. We use a mean of observed λ values over iterations. Finally, the $\hat{\alpha}_{n+1}$ is tuned based on $\hat{\lambda}_n$. We use a linear weighting update rule: $\hat{\alpha}_{n+1} = t\hat{\lambda}_n\hat{\alpha}_n + (1-t)\hat{\alpha}_n$ for $t \in (0, 1)$, where $\hat{\lambda}_n\hat{\alpha}_n$ is a conservative estimate of β.

8 Empirical Evaluation

We study the empirical properties of DAGGER, Imitation Gradient (IG) and Multiple Imitation Gradient (MIG), showing that even in low dimensional and convex settings, the implications of the convergence of policies become apparent. We intentionally sought out cases and chose parameters such that these on-policy algorithms do not achieve convergence in order to better understand their properties. We consider completely deterministic domains simply for the sake of accurately measuring the true dynamic regret. In this evaluation, we attempt to address the following questions: (1) How are policy performance and dynamic regret affected by changing system parameters? (2) Can Adaptive On-Policy Regularization improve convergence of the average dynamic regret and policy performance?

8.1 Cart-Pole Balancing

First we consider a task where the robot learns to push a cart left or right with a fixed force magnitude in order to balance a pole upright over 100 iterations. The control space is discrete $\{\text{left}, \text{right}\}$ and the the state space consists of cart location and velocity and pole angle and angular velocity. We measure the absolute performance of a policy as the angular deviation from the upright position. We obtained a nonlinear algorithmic supervisor via reinforcement learning. The robot's policy was learned using l_2-regularized least squares as in (4). In this setting, we vary the difficulty of the problem, i.e. controlling β, by setting the force magnitude to either low or high values corresponding to easy and hard settings, respectively. For all algorithms, the regularization was initially set to $\hat{\alpha}_1 = 0.1$. Stepsizes η for IG and MIG were set to 0.0001 and 0.01, respectively. Further details can be found in the appendix.

The instantaneous regrets for all three imitation learning algorithms, measured as $f_n(\theta_n) - \min_\theta f_n(\theta)$, are shown in the left column of the top and middle rows of Fig. 1 for both the easy and hard settings, respectively. In the right column are the actual angular deviations of the pole. Because the convergence of static regret is guaranteed regardless of the difficulty, we omit it for clarity. In the easy setting, the algorithms converge to costs similar to the supervisor. In the hard setting, there is chattering during the learning process. We observe that regret does not always converge indicating a discrepancy between the supervisor and the learner, which is consistent with the theoretical results that suggest difficulty influences convergence to the best policy.

The bottom row of Fig. 1 shows that Adaptive On-Policy Regularization (AOR) can be used to improve dynamic regret. In our implementation of AOR, we set $t = 0.01$ and updated $\hat{\lambda}_n$ and $\hat{\alpha}_n$ every ten iterations. For IG and MIG with AOR, we adjusted the stepsize η as a function of $\hat{\alpha}_n$ as motivated by the conditions in the theorems. In practice this counteracts exorbitantly large gradients caused by large $\hat{\alpha}$ values. DAGGER without AOR exhibits severe chattering. With AOR, the performance is stabilized, leading to a converged policy. A similar result is observed for MIG. We note that IG did not exhibit chattering and the

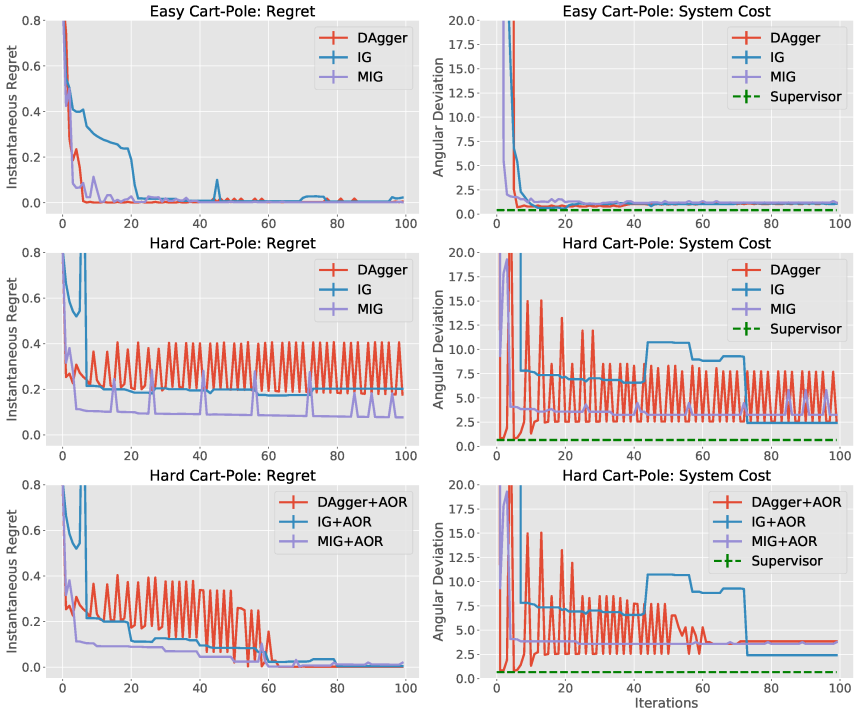

Fig. 1: Instantaneous regrets are shown in the right column while actual system cost measured as angular deviation are shown in the left column. All three on-policy algorithms without adaptive regularization are shown on both the easy (top row) and hard (middle and bottom rows) versions of cart-pole. This empirical result shows existence of a case where the system is difficult enough that the algorithms suffer unstable learning curves. The average dynamic regret does not converge. The bottom row shows the algorithms on the hard setting but using AOR. The dynamic regret tends towards zero and the chattering in DAGGER and MIG is reduced.

learning curve was entirely unaffected. We attribute this to the discrete nature of the control space; even if the policy parameters have different regret rates, the resulting trajectories could be the same. We also evaluated the same task over several different initial conditions. Since averaging the results tends to hide the chattering, individual plots are give in the appendix.

8.2 Walker Locomotion

Next, we consider a 2-dimensional walker from the OpenAI Gym, where the objective is to move the farthest distance. Again we induced difficulty and sub-optimal policies by increasing the force magnitude of controls. Additional information on the environment is in the appendix. Ridge regression was used for the robot's policy. Here, we set $\alpha_1 = 1.0$ and $t = 0.1$ for AOR. We used the same initial η values for IG and MIG and adjusted the stepsize as a function of $\hat{\alpha}_n$ when using AOR. The results are shown in Fig. 2. Without adaptive regularization, the average dynamic regret fails to converge and all distance curves exhibit severe chattering with no stable learning. With AOR, average dynamic regret converges to zero and all distance curves are stabilized.

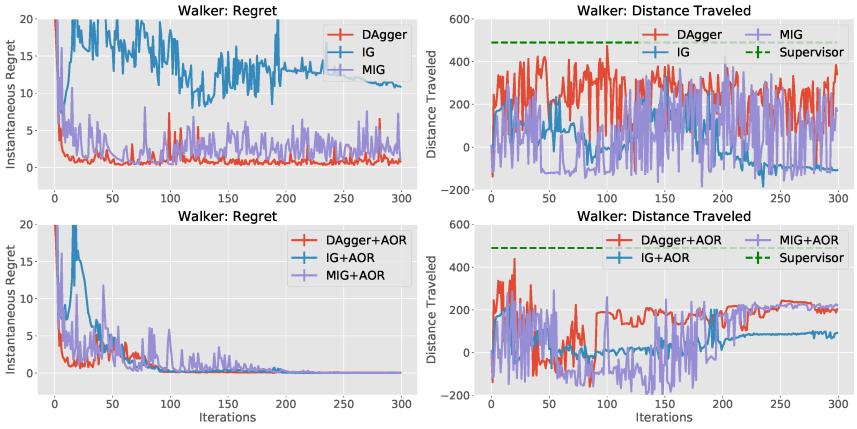

Fig. 2: The instantaneous regrets fail to converge to zero without AOR (top row) and the distance curves exhibit chattering. With AOR (bottom row), the average dynamic regret converges and chattering is reduced after 200 iterations.

9 Discussion and Future Work

Dynamic regret theory offers a promising method for theoretical analysis in imitation learning. The question of whether an online algorithm leads to converged or stable policies is inherently captured in dynamic regret, in contrast to static regret which captures policy performance on the average of the distributions. Theoretical analyses suggest new conditions to guarantee convergence. The simulation results suggest that the questions of convergence and optimality must be addressed when designing an imitation learning robotics systems because stable performance is not always guaranteed. Indeed, even if static regret is found to be low, the distributions induced by the robot during training may be far too unpredictable for stable policy performance to actually be achieved.

In this paper, the analyses relied on Assumption 4. By modeling the problem with this assumption, we constrained the dynamics to aid the analysis. If other properties were known about the dynamics, then additional information could improve regret rates and conditions for optimality. For example, similar problem statements were studied in [5] and [12] in general Online Optimization. It was recently shown that augmenting the mostly model-free analyses with model-based learning can improve static regret bounds and performance [3]. We hypothesize that dynamic regret rates can also be improved in this way.

Acknowledgments

This research was performed at the AUTOLAB at UC Berkeley in affiliation with the Berkeley AI Research (BAIR) Lab, Berkeley Deep Drive (BDD), the Real-Time Intelligent Secure Execution (RISE) Lab, and the CITRIS "People and Robots" (CPAR) Initiative and by the Scalable Collaborative Human-Robot Learning (SCHooL) Project, NSF National Robotics Initiative Award 1734633. The authors were supported in part by donations from Siemens, Google, Amazon Robotics, Toyota Research Institute, Autodesk, ABB, Samsung, Knapp, Loccioni, Honda, Intel, Comcast, Cisco, and Hewlett-Packard. We thank the reviewers for their valuable comments and also our colleagues, in particular Jeffrey Mahler, Ching-An Cheng, and Brijen Thananjeyan for their insights.

References

1. J Andrew Bagnell. An invitation to imitation. Technical report, Carnegie Mellon Univ Pittsburgh PA Robotics Inst, 2015.
2. Ching-An Cheng and Byron Boots. Convergence of value aggregation for imitation learning. *International Conference on Artificial Intelligence and Statistics*, 2018.
3. Ching-An Cheng, Xinyan Yan, Evangelos Theodorou, and Byron Boots. Model-based imitation learning with accelerated convergence. *arXiv preprint arXiv:1806.04642*, 2018.
4. Ching-An Cheng, Xinyan Yan, Nolan Wagener, and Byron Boots. Fast policy learning through imitation and reinforcement. In *Conference on Uncertainty in Artificial Intelligence*, 2018.
5. Eric C Hall and Rebecca M Willett. Online convex optimization in dynamic environments. *IEEE Journal of Selected Topics in Signal Processing*, 9(4):647–662, 2015.
6. Elad Hazan. Introduction to online convex optimization. *Foundations and Trends in Optimization*, 2(3-4):157–325, 2016.
7. Elad Hazan and Satyen Kale. Beyond the regret minimization barrier: optimal algorithms for stochastic strongly-convex optimization. *The Journal of Machine Learning Research*, 15(1):2489–2512, 2014.
8. Elad Hazan and Comandur Seshadhri. Adaptive algorithms for online decision problems. *Electronic colloquium on computational complexity (ECCC)*, 2007.
9. Michael Laskey, Caleb Chuck, Jonathan Lee, Jeffrey Mahler, Sanjay Krishnan, Kevin Jamieson, Anca Dragan, and Ken Goldberg. Comparing human-centric and robot-centric sampling for robot deep learning from demonstrations. In *IEEE International Conference on Robotics and Automation (ICRA), 2017*, 2017.
10. Aryan Mokhtari, Shahin Shahrampour, Ali Jadbabaie, and Alejandro Ribeiro. Online optimization in dynamic environments: Improved regret rates for strongly convex problems. In *Decision and Control (CDC), 2016 IEEE 55th Conference on*, pages 7195–7201. IEEE, 2016.
11. Takayuki Osa, Joni Pajarinen, Gerhard Neumann, J Andrew Bagnell, Pieter Abbeel, Jan Peters, et al. An algorithmic perspective on imitation learning. *Foundations and Trends in Robotics*, 7(1-2):1–179, 2018.
12. Sasha Rakhlin and Karthik Sridharan. Optimization, learning, and games with predictable sequences. In *Advances in Neural Information Processing Systems*, pages 3066–3074, 2013.
13. Stéphane Ross, Geoffrey J Gordon, and J Andrew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. *International Conference on Artificial Intelligence and Statistics*, 2011.
14. Wen Sun, Arun Venkatraman, Geoffrey J Gordon, Byron Boots, and J Andrew Bagnell. Deeply aggrevated: Differentiable imitation learning for sequential prediction. In *International Conference on Machine Learning*, 2017.
15. Tianbao Yang, Lijun Zhang, Rong Jin, and Jinfeng Yi. Tracking slowly moving clairvoyant: Optimal dynamic regret of online learning with true and noisy gradient. In *International Conference on Machine Learning*, 2016.
16. Lijun Zhang, Tianbao Yang, Jinfeng Yi, Jing Rong, and Zhi-Hua Zhou. Improved dynamic regret for non-degenerate functions. In *Advances in Neural Information Processing Systems*, 2017.
17. Martin Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *Proceedings of the 20th International Conference on Machine Learning (ICML-03)*, pages 928–936, 2003.

10 Appendix

For convenience, the main assumptions of the paper are reproduced here:

- For all $\theta_1, \theta_2, \theta \in \Theta$, $\exists \alpha > 0$ such that

$$f_\theta(\theta_2) \geq f_\theta(\theta_1) + \langle \nabla f_\theta(\theta_1), \theta_2 - \theta_1 \rangle + \frac{\alpha}{2}\|\theta_1 - \theta_2\|^2.$$

- For all $\theta_1, \theta_2, \theta \in \Theta$, $\exists \gamma > 0$ such that

$$\|\nabla f_\theta(\theta_1) - \nabla f_\theta(\theta_2)\| \leq \gamma \|\theta_1 - \theta_2\|$$

  and $\exists G > 0$ such that $\|\nabla f_\theta(\theta_1)\| \leq G$.
- For all $\theta' \in \Theta$, θ^* is in the relative interior of Θ where $\theta^* = \arg\min_{\theta \in \Theta} f_{\theta'}(\theta)$. That is, $\nabla f_{\theta'}(\theta^*) = 0$.
- For all $\theta_1, \theta_2, \theta \in \Theta$, $\exists \beta > 0$ such that

$$\|\nabla f_{\theta_1}(\theta) - \nabla f_{\theta_2}(\theta)\| \leq \beta \|\theta_1 - \theta_2\|.$$

The proofs in this paper will make heavy use of the following well-known result on the strong convexity of a function [7].

Lemma 2. *The following holds for all $\theta \in \Theta$ and $\theta^* = \arg\min_{\theta'} f(\theta')$:*

$$f(\theta) - f(\theta^*) \geq \frac{\alpha}{2}\|\theta - \theta^*\|^2.$$

10.1 Proof of Proposition 1

Proof. By strong convexity of f_m, we have

$$\begin{aligned}\frac{\alpha}{2}\|\theta_m^* - \theta_n^*\|^2 &\leq f_m(\theta_n^*) - f_m(\theta_m^*) \\ &\leq \|\nabla f_m(\theta_n^*)\|\|\theta_n^* - \theta_m^*\| - \frac{\alpha}{2}\|\theta_m^* - \theta_n^*\|^2\end{aligned}$$

Then by rearranging terms, $\|\theta_m^* - \theta_n^*\| \leq \frac{1}{\alpha}\|\nabla f_m(\theta_n^*) - \nabla f_n(\theta_n^*)\| \leq \frac{\beta}{\alpha}\|\theta_m - \theta_n\|$, where the last inequality uses Assumption 4. □

10.2 Proof of Theorem 1

Before directly proving this theorem, we establish several supporting results based on the path variation.

Lemma 3. *For a sequence of predictions made by the online gradient descent algorithm $\theta_{1:N}$ and a sequence of optimal parameters $\theta_{1:N}^*$, the following inequality holds on the path variation:*

$$V(\theta_{1:N}^*) \leq \eta \frac{\beta\gamma}{\alpha} \sum_{n=1}^{N} \|\theta_n - \theta_n^*\|.$$

Proof. From Proposition 1, we have $\|\theta^*_{n+1} - \theta^*_n\| \leq \frac{\beta}{\alpha}\|\theta_{n+1} - \theta_n\| = \frac{\beta}{\alpha}\|\eta\nabla f_n(\theta_n)\| = \eta\frac{\beta}{\alpha}\|\nabla f_n(\theta_n) - \nabla f_n(\theta^*_n)\| \leq \eta\frac{\beta\gamma}{\alpha}\|\theta_n - \theta^*_n\|$, where the the final equality uses Assumption 3 and the final inequality uses Assumption 2. Then the result follows immediately. □

Lemma 4. *Let $\rho = (1 - \alpha\eta + \gamma^2\eta^2)^{1/2}$, which is always non-negative for any positive choice of η because $\gamma \geq \alpha$ by definition. Then the following inequality holds*

$$\sum_{n=1}^{N} \|\theta_n - \theta^*_n\| \leq \|\theta_1 - \theta^*_1\| + \sum_{n=1}^{N} \rho\|\theta_n - \theta^*_n\| + V(\theta^*_{1:N}).$$

Proof. By strong convexity we have the following: $0 \leq 2(f_n(\theta_n) - f_n(\theta^*_n)) \leq 2\langle\nabla f_n(\theta_n), \theta_n - \theta^*_n\rangle - \alpha\|\theta^*_n - \theta_n\|^2$. By the update rule given in the algorithm:

$$\begin{aligned}\|\theta_{n+1} - \theta^*_n\|^2 &= \|\theta_n - \eta\nabla f_n(\theta_n) - \theta^*_n\|^2 \\ &= \|\eta\nabla f_n(\theta_n)\|^2 + \|\theta_n - \theta^*_n\|^2 - 2\eta\langle\nabla f_n(\theta_n), \theta_n - \theta^*_n\rangle. \quad (6)\end{aligned}$$

By rearranging the terms in (6) and combining with the very first inequality, we arrive at the following:

$$\|\theta_{n+1} - \theta^*_n\|^2 \leq (1 - \alpha\eta)\|\theta_n - \theta^*_n\|^2 + \|\eta\nabla f_n(\theta_n)\|^2.$$

Using Assumption 2 and the fact that $\nabla f_n(\theta^*_n) = 0$ and the smoothness of f_n:

$$\begin{aligned}\|\theta_{n+1} - \theta^*_n\|^2 &\leq \|\theta_n - \theta^*_n\|^2 - \alpha\eta\|\theta_n - \theta^*_n\|^2 + \eta^2\|\nabla f_n(\theta_n) - \nabla f_n(\theta^*_n)\|^2 \\ &\leq \left(1 - \alpha\eta + \gamma^2\eta^2\right)\|\theta_n - \theta^*_n\|^2. \quad (7)\end{aligned}$$

Then let $\rho = \left(1 - \alpha\eta + \gamma^2\eta^2\right)^{1/2}$. Following from [10], consider the series:

$$\begin{aligned}\sum_{n=1}^{N} \|\theta_n - \theta^*_n\| &= \|\theta_1 - \theta^*_1\| + \sum_{n=2}^{N} \|\theta_n - \theta^*_{n-1} + \theta^*_{n-1} - \theta^*_n\| \\ &\leq \|\theta_1 - \theta^*_1\| + \sum_{n=2}^{N} \|\theta_n - \theta^*_{n-1}\| + V(\theta^*_{1:N}) \\ &\leq \|\theta_1 - \theta^*_1\| + \sum_{n=1}^{N} \rho\|\theta_n - \theta^*_n\| + V(\theta^*_{1:N}),\end{aligned}$$

where the second line uses the definition of the path variation and the third line uses (7). □

Proof (Proof of Theorem 1). We begin by bounding the result from Lemma 4 above by Lemma 3:

$$\sum_{n=1}^{N} \|\theta_n - \theta^*_n\| \leq \|\theta_1 - \theta^*_1\| + \left(\rho + \eta\frac{\beta\gamma}{\alpha}\right)\sum_{n=1}^{N} \|\theta_n - \theta^*_n\|.$$

By rearranging the terms and bounding by the diameter of Θ:

$$\sum_{n=1}^{N} \|\theta_n - \theta_n^*\| \leq \frac{D}{1 - \rho - \eta\frac{\beta\gamma}{\alpha}}. \tag{8}$$

It can be shown that, under the assumptions, the choice of $\eta = \frac{\alpha(\alpha^2 - 2\beta\gamma)}{2\gamma^2(\alpha^2 - \beta^2)}$ ensures that $\left(1 - \rho - \eta\frac{\beta\gamma}{\alpha}\right)$ is positive. By the G-Lipschitz continuity of f_n, we have

$$\sum_{n=1}^{N} f_n(\theta_n) - \sum_{n=1}^{N} f_n(\theta_n^*) \leq \frac{GD}{1 - \rho - \eta\frac{\beta\gamma}{\alpha}},$$

and so $R_D(\theta_1, \ldots, \theta_N) = O(1)$. So we have $\frac{1}{N}R_D = O(1/N)$ which goes to zero.

Convergence of $\|\theta_n - \theta_n^*\|$ to zero can proved in the following way. Because we bounded the dynamic regret by a constant, we have that there exists some non-negative constant B such that $\lim_{N\to\infty} \sum_{n=1}^{N} f_n(\theta_n) - f_n(\theta_n^*) \leq B$. From Lemma 2, we have $\|\theta_n - \theta_n^*\|^2 \leq \frac{2}{\alpha}\left(f_n(\theta_n) - f_n(\theta_n^*)\right)$, and so the following inequality holds:

$$\sum_{n=1}^{\infty} \|\theta_n - \theta_n^*\|^2 \leq \frac{2B}{\alpha}.$$

Define $\alpha_n = \|\theta_n - \theta_n^*\|^2$. Also define the partial series $p_N := \sum_{n=1}^{N} a_n$. Note that the sequence $(p_n)_{n=1}^{N}$ is monotonic because $a_n \geq 0$. By the monotone convergence theorem, we have that there is a non-negative limit point L such that $\lim_{N\to\infty} p_N = L$. L is the supremum of $(p_n)_{n=1}^{N}$. We apply a known result from analysis that if an infinite series converges, its sequence of elements converges to zero. Because $\lim_{N\to\infty}(p_n)_{n=1}^{N}$ converges and $\lim_{N\to\infty}(a_n)_{n=1}^{N}$ is the sequence of elements, then $\lim_{N\to\infty} a_n = 0$. So by definition of a_n, the sequence $(\|\theta_n - \theta_n^*\|)_{n\in\mathbb{N}}$ converges to zero. This tells us that the policy parameters converge to the optima.

The proof of convergence of the parameters takes a similar approach. We note that from the proof of Lemma 3, we have

$$\frac{1}{\eta\gamma}\|\theta_{n+1} - \theta_n\| \leq \|\theta_n - \theta_n^*\|,$$

which means if we define $b_n := \|\theta_{n+1} - \theta_n\|$ and $q_N := \sum_{n=1}^{N} b_n$, we have $\lim_{N\to\infty} q_N \leq \eta\gamma\frac{D}{1-\rho-\eta\frac{\beta\gamma}{\alpha}}$ for all N. Since the series of distances between consecutive elements converges $(b_n)_{n\in\mathbb{N}}$, we have that the sequence $(\theta_n)_{n=1}^{N}$ converges in the limit.

□

10.3 Proof of Lemma 1

The result of Lemma 1 is a slight modification from Theorem 3 from Zhang et al. [16], which is reproduced in full here with supporting lemmas for completeness.

For all following proofs, unless stated otherwise, we assume $f : \Theta \mapsto \mathbb{R}$ always satisfies Assumptions 1-3. Again before proving this theorem, we establish a crucial lemma. As in the proof of Theorem 1, we show a bound on the improvement from a single gradient step.

Lemma 5. *Let θ' be the current parameter played by the algorithm at any iteration, $\hat{\theta} = P_\Theta(\theta' - \eta \nabla f_n(\theta'))$ and $\theta_n^* = \arg\min_\theta f_n(\theta)$. Then we have*

$$\|\hat{\theta} - \theta_n^*\|^2 \leq \left(1 - \frac{2\alpha}{1/\eta + \alpha}\right) \|\theta' - \theta_n^*\|^2.$$

Proof. By the update rule:

$$\begin{aligned}
\hat{\theta} &= P_\Theta(\theta' - \eta \nabla f_n(\theta')) \\
&= \arg\min_{\theta \in \Theta} \|\theta' - \eta \nabla f_n(\theta') - \theta\|^2 \\
&= \arg\min_{\theta \in \Theta} \left\{ 2\langle \eta \nabla f_n(\theta'), \theta - \theta' \rangle + \|\theta' - \theta\|^2 + \|\eta^2 \nabla f_n(\theta')\|^2 \right\} \\
&= \arg\min_{\theta \in \Theta} \left\{ f_n(\theta') + \langle \nabla f_n(\theta'), \theta - \theta' \rangle + \frac{1}{2\eta} \|\theta' - \theta\|^2 \right\} \\
&= \arg\min_{\theta \in \Theta} h_n(\theta)
\end{aligned}$$

where we define $h_n(\theta) := f_n(\theta') + \langle \nabla f_n(\theta'), \theta - \theta' \rangle + \frac{1}{2\eta} \|\theta' - \theta\|^2$. Note that $h(\theta)$ is $\frac{1}{\eta}$-strongly convex. So by applying Lemma 2 to h_n and by the fact that $\hat{\theta}$ is the minimizer of h, we have

$$h_n(\hat{\theta}) \leq h_n(\theta_n^*) - \frac{1}{2\eta} \|\hat{\theta} - \theta_n^*\|^2$$

By the strong convexity of f_n it holds that $f_n(\theta') + \langle \nabla f_n(\theta'), \theta_n^* - \theta' \rangle \leq f_n(\theta_n^*) - \frac{\alpha}{2} \|\theta' - \theta_n^*\|^2$. By smoothness and the fact that $\eta < 1/\gamma$ and Θ is Euclidean, we also have

$$f_n(\hat{\theta}) \leq f_n(\theta') + \langle \nabla f_n(\theta'), \hat{\theta} - \theta' \rangle + \frac{\gamma}{2} \|\theta' - \hat{\theta}\|^2 \leq h_n(\hat{\theta})$$

Combining these inequalities gives

$$f_n(\hat{\theta}) \leq f_n(\theta_n^*) - \frac{\alpha}{2} \|\theta' - \theta_n^*\|^2 + \frac{1}{2\eta} \|\theta' - \theta_n^*\|^2 - \frac{1}{2\eta} \|\hat{\theta} - \theta_n^*\|^2$$

By applying Lemma 2 again we have:

$$\frac{\alpha}{2} \|\hat{\theta} - \theta_n^*\|^2 \leq -\frac{\alpha}{2} \|\theta' - \theta_n^*\|^2 + \frac{1}{2\eta} \|\theta' - \theta_n^*\|^2 - \frac{1}{2\eta} \|\hat{\theta} - \theta_n^*\|^2$$

The result can be obtained by rearranging and aggregating the terms and then simplifying. □

We now present the proof of Lemma 1.

Proof (Proof of Lemma 1). According to the update rule of multiple gradients we apply the result established in Lemma 5 K times which gives

$$\begin{aligned}\|\theta_{n+1} - \theta_n^*\|^2 &\leq \left(1 - \frac{2\alpha}{1/\eta + \alpha}\right)^K \|\theta_n - \theta_n^*\|^2 \\ &\leq \exp\left(-\frac{2\alpha K}{1/\eta + \alpha}\right) \|\theta_n - \theta_n^*\|^2 \\ &\leq \frac{1}{4}\|\theta_n - \theta_n^*\|^2\end{aligned}$$

Then, as in Theorem 1, we bound the distances between the optimal parameters and the algorithm's parameters.

$$\begin{aligned}\sum_{n=1}^{N} \|\theta_n - \theta_n^*\|^2 &= \|\theta_1 - \theta_1^*\|^2 + \sum_{n=2}^{N} \|\theta_n - \theta_{n-1}^* + \theta_{n-1}^* - \theta_n^*\|^2 \\ &\leq \|\theta_1 - \theta_1^*\|^2 + 2\sum_{n=2}^{N} \|\theta_n - \theta_{n-1}^*\|^2 + \|\theta_{n-1}^* - \theta_n^*\|^2 \\ &\leq \|\theta_1 - \theta_1^*\|^2 + \sum_{n=2}^{N} \frac{1}{2}\|\theta_{n-1} - \theta_{n-1}^*\|^2 + 2\|\theta_{n-1}^* - \theta_n^*\|^2 \\ &\leq 2\|\theta_1 - \theta_1^*\|^2 + 4\sum_{n=2}^{N} \|\theta_{n-1}^* - \theta_n^*\|^2 \\ &\leq 2\|\theta_1 - \theta_1^*\|^2 + 4S\left(\theta_{1:N}^*\right)\end{aligned}$$

Finally, by the smoothness of all f_n we have

$$\begin{aligned}\sum_{n=1}^{N} f_n(\theta_n) - f_n(\theta_n^*) &\leq \sum_{n=1}^{N} \frac{\gamma}{2}\|\theta_n - \theta_n^*\|^2 \\ &\leq 2\gamma S(\theta_{1:N}^*) + \gamma\|\theta_1 - \theta_1^*\|^2.\end{aligned}$$

□

10.4 Full Proof of Theorem 2

We begin by upper bounding the squared path variation using Proposition 1 and Assumption 4:

$$\begin{aligned}\|\theta^*_{n+1} - \theta^*_n\| &\leq \frac{\beta}{\alpha}\|\theta_{n+1} - \theta_n\| \\ &\leq \frac{\beta\eta}{\alpha}\|\nabla f_n(\theta^1_n) + \ldots + \nabla f_n(\theta^K_n)\| \\ &\leq \frac{\beta\eta}{\alpha}\sum_{j=1}^{K}\|\nabla f_n(\theta^j_n)\| \\ &= \frac{\beta\eta}{\alpha}\sum_{j=1}^{K}\|\nabla f_n(\theta^j_n) - \nabla f_n(\theta^*_n)\|\end{aligned}$$

Next we apply Lemma 5 as before:

$$\begin{aligned}\|\theta^*_{n+1} - \theta^*_n\| &\leq \frac{\beta\eta\gamma}{\alpha}\sum_{j=1}^{K}\|\theta^j_n - \theta^*_n\| \\ &\leq \frac{\beta\eta\gamma}{\alpha}\|\theta_n - \theta^*_n\|\sum_{j=1}^{K}\left(1 - \frac{2\alpha}{1/\eta + \alpha}\right)^j \\ &\leq \frac{\beta\eta\gamma K}{\alpha}\|\theta_n - \theta^*_n\|\end{aligned}$$

Summing over all n, we have

$$S(\theta^*_{1:N}) \leq \left(\frac{\beta\eta\gamma K}{\alpha}\right)^2 \sum_{n=1}^{N}\|\theta_n - \theta^*_n\|^2$$

Then by substituting into Lemma 1 and bounding the quantity from above using the strong convexity of f_n, we have

$$\begin{aligned}R_D(\theta_1, \ldots \theta_N) &\leq \gamma\|\theta_1 - \theta^*_1\|^2 + \frac{2\beta^2\eta^2\gamma^3K^2}{\alpha^2}\sum_{n=1}^{N}\|\theta_n - \theta^*_n\|^2 \\ &\leq \gamma\|\theta_1 - \theta^*_1\|^2 + \frac{4\beta^2\eta^2\gamma^3K^2}{\alpha^3}\sum_{n=1}^{N} f_n(\theta_n) - f_n(\theta^*_n) \\ &\leq \frac{\gamma\|\theta_1 - \theta^*_1\|^2}{1 - \frac{4\beta^2\eta^2\gamma^3K^2}{\alpha^3}} \leq \frac{\gamma D^2}{1 - \frac{4\beta^2\eta^2\gamma^3K^2}{\alpha^3}}\end{aligned}$$

It can be verified that for $\eta < \min\left\{1/\gamma, \frac{\alpha^{5/2} - \gamma^{3/2}\beta\log 4}{2\gamma^{3/2}\alpha\beta\log 4}\right\}$ and $K > \frac{1/\eta + \alpha}{2\alpha}\log 4$, the denominator is positive.

The proof of convergence of the $\|\theta_n - \theta_n^*\|$ to zero is identical to that of Theorem 1.

The proof of convergence of the sequence $\theta_{1:N}$ similar, but we must return to the path variation. We begin again with the proof of Lemma 1 where we have for any n

$$\|\theta_{n+1} - \theta_n^*\| \leq \frac{1}{2}\|\theta_n - \theta_n^*\|. \tag{9}$$

Then, as before, we bound the sum of the differences between policy parameters and optima, but without the square

$$\sum_{n=1}^{N} \|\theta_n - \theta_n^*\| \leq 2\|\theta_1 - \theta_1^*\| + 2V(\theta_{1:N}^*) \tag{10}$$

By bounding the path variation from above using Lemma 5 and again rearranging terms, a constant upper bound on $\sum_{n=1}^{N} \|\theta_n - \theta_n^*\|$ is established in the same way as Theorem 1 except we require $1 > \frac{2\beta\eta\gamma K}{\alpha}$, but this is satisfied with the same condition on η. The rest of proof proceeds exactly in the same way as Theorem 1. Therefore with Multiple Imitation Gradient, $\theta_{1:N}$ converges in the limit.

10.5 Additional Information and Experiments on Cart-Pole

The code is available at https://github.com/jon--lee/aopr.

To generate the easy and hard versions of cart-pole balancing, we varied the parameter controlling the force magnitude applied with each left or right control. A low force magnitude of 2.0 was used for the easy setting and a higher force magnitude of 10.0 was used for the hard setting. The value 2.0 was the smallest integer before the force was too low to control the cart. The value 10.0 was one of the highest before we noticed the average λ values began to decrease as a function of the force magnitude. As mentioned, the parameters η and α_1 where intentionally chosen so that the task would exhibit unstable or suboptimal results. Trajectories were 200 time steps long.

In order to estimate $\hat{\lambda}_n$ between each iteration, we compute $\|\theta_n - \theta_{n-1}\|$ directly since both quantities are known. Because each f_n are strongly convex supervised learning problems, full information is known f_n and so the minimum $\theta_n^* = \arg\min f_n(\theta)$ can be solved. In this case, f_n corresponded to a l_2-regularized ridge regression problem which has a closed form solution. In the case of stochastic problems, it may be necessary to obtain a sample estimate of f_n first by collecting several trajectories per iteration and then estimate θ_n^* from the sample average. Our estimate of $\hat{\lambda}_n$ was simply the ratio of these normed differences averaged over iterations. Note that λ as defined in Assumption 4, is a global constant, but in practice only local regions may be relevant for the problem at hand, which is why we estimate λ at each iteration.

Here, we run the same cart-pole experiment but over 10 different initial pole angle conditions. The cost curves are shown, measured as angular deviation from

the upright position. The top row show shows without AOR and the bottom row shows with AOR. Each column corresponds to the same initial conditions.

To conserve space, only the last 50 iterations are shown, which is when the curves are typically stabilized by adaptive regularization. DAGGER and MIG see a reduction of chattering in most cases when using AOR in Fig. 3 and Fig. 4, while IG in Fig. 5, as before, has no difference at all.

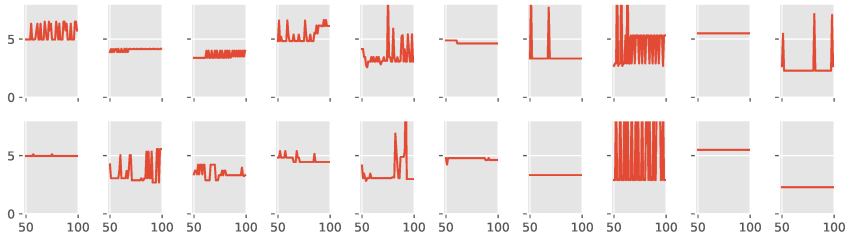

Fig. 3: DAGGER angular deviations over 10 different initial conditions.

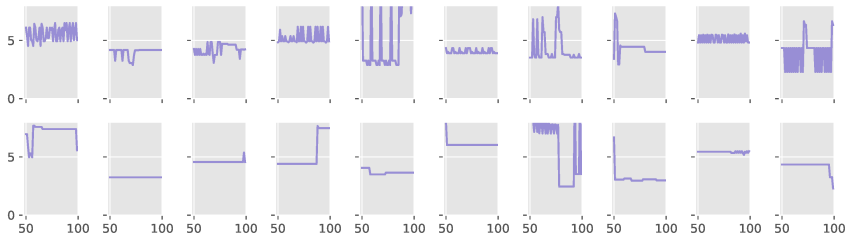

Fig. 4: MIG angular deviations over 10 different initial conditions.

10.6 Additional Information and Experiments in Walker

To induce high regret policies in the walker domain, we increased the force of controls five-fold. As in cart-pole, trajectories consisted of 200 time steps and one trajectory was collected and evaluated at each of the 300 iterations. Again, there was no stochasticity in the environment for the sake of computing the instantaneous regret.

We also evaluated the effect of AOR on a different OpenAI gym task hopper with exactly the same hyperparameter settings. The results are shown in Fig. 6. We note that although chattering is reduced, the distance traveled on average suffers. This is one disadvantage of adaptive regularization that was noted earlier. Increased regularization, although leading to convergence of average dynamic

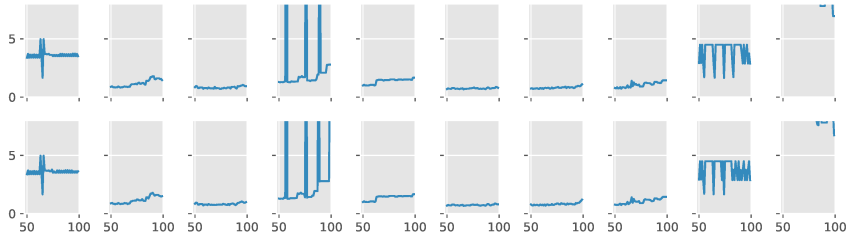

Fig. 5: IG angular deviations over 10 different initial conditions.

regret, can cause poor policy performance if it is excessive. This example suggests that from a practical perspective, it is important to monitor both the regret and whatever qualitative or quantitative metrics are actually desired to ensure that they agree.

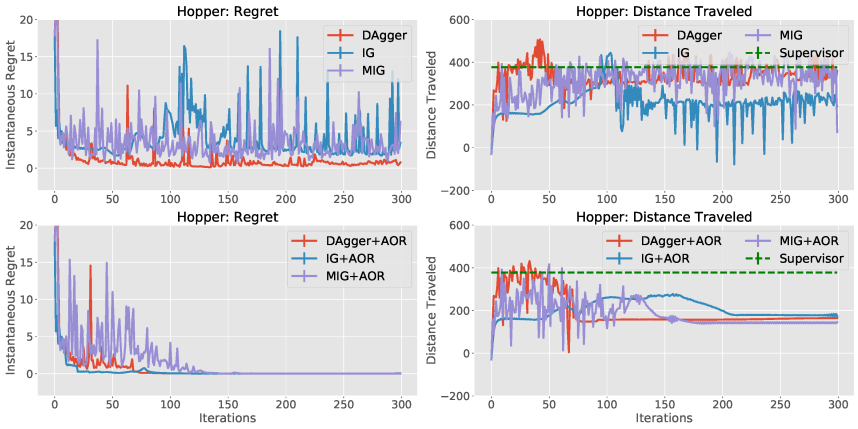

Fig. 6: OpenAI Gym hopper on the same conditions as the walker experiment. As observed in prior experiments, on-policy algorithms without proper regularization lead to unstable learning. With adaptive regularization the learning is stabilized; however, in this case, performance on the system is reduced.

10.7 Discussion on the Relation Between Regularization and Stepsize

Both the IG and MIG algorithms use an additional hyperparameter as part of the optimization procedure: the stepsize η. As the regularization therm α is increased, whether using AOR or just arbitrarily, the stepsize in theory should decrease based on the conditions presented in Theorem 1 and Theorem 2. Therefore, there is an inherent connection between the stepsize and the amount of

regularization. This is similar to results in convex optimization: increased strong convexity parameters imply increased smoothness parameters because $\alpha < \gamma$. This, in turn, calls for small stepsizes to achieve converges guarantees.

We observe the same effect in MIG and IG, especially when using AOR. High regularization means better convergence and low regret. At each iteration, we want the update to take us in the direction of the optimal parameter. But the step should be small enough such that at the next iteration, the optimal parameter does not move too far. Therefore, the stepsize must be carefully balanced. This motivates the adaptive updates to η in AOR as well. η can be thought of as a function of α rather than a separate, independent hyperparameter. For reference, the change in regularization of $\hat{\alpha}_n$ on all three domains is give in Fig. 7.

10.8 Empirical Results with Static Regret

It has been proven theoretically that static regret converges to zero under the assumptions in this paper and these experiments [6,13]. In this section, we empirically evaluate these results for completeness. The results compare the static regret of all three algorithms with and without AOR across and the actual system cost is reproduced side-by-side for convenience.

As shown in Fig. 8, Fig. 9, and Fig. 10, static regret nearly always tends to zero regardless of the actual policy performance or chattering. This is consistent with the theoretical results of prior work [6].

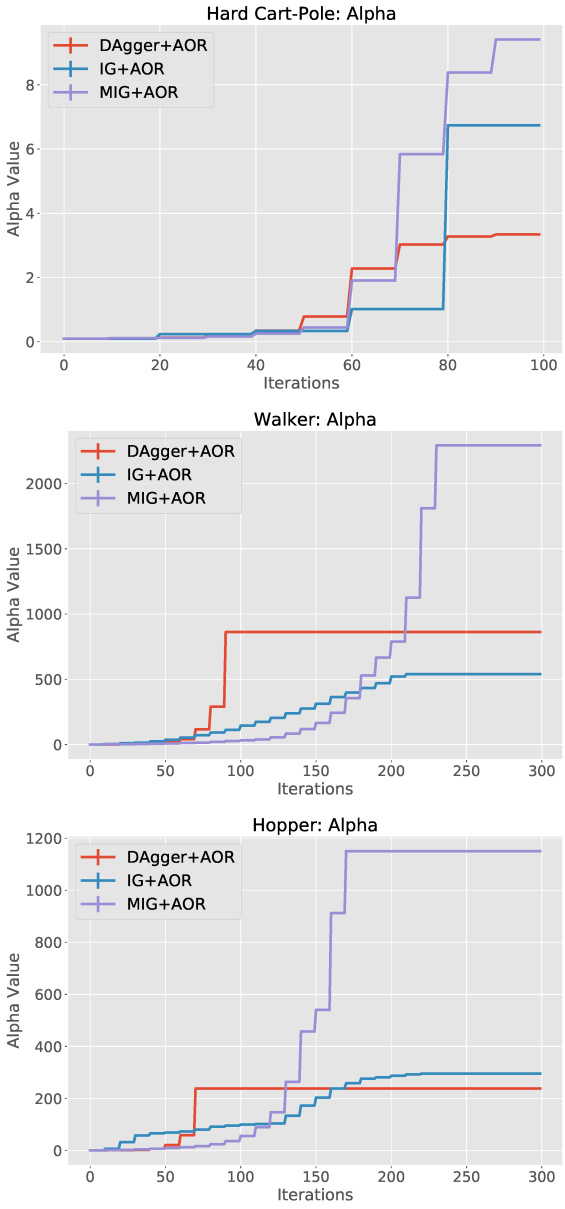

Fig. 7: The regularization parameter $\hat{\alpha}_n$ using AOR increases until the instantaneous regret converges in all three experimental domains. In Walker and Hopper, especially high regularization is needed to guarantee convergence in regret.

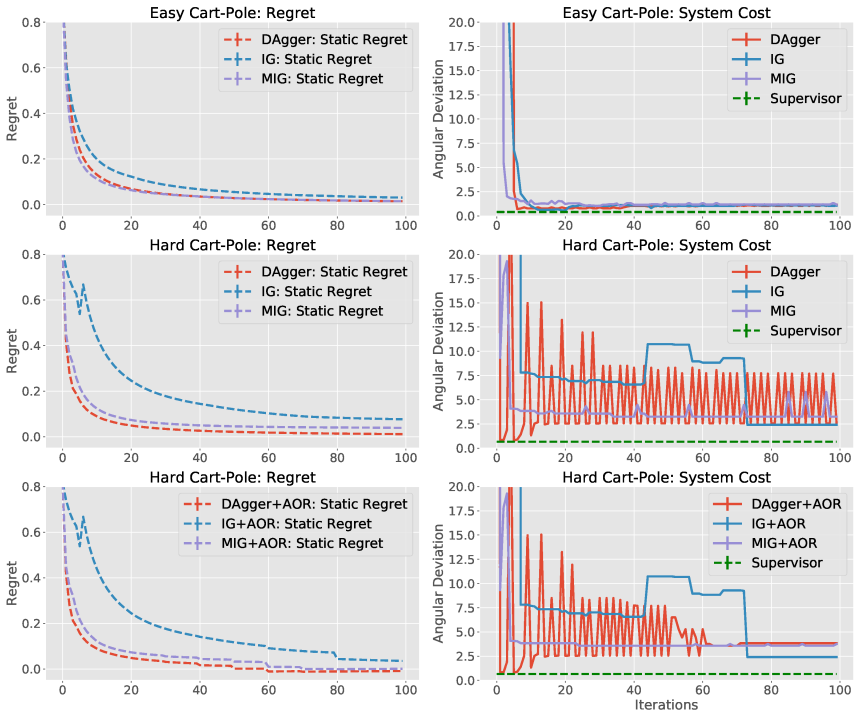

Fig. 8: The identical cart-pole experiment as in Fig. 1 is reproduced but plotting static regret instead of the instantaneous regret. In this environment, the static regret quickly decreases to zero with and without AOR and even in the difficult domain. The regret shows no indication that the actual policy is performing poorly over iterations.

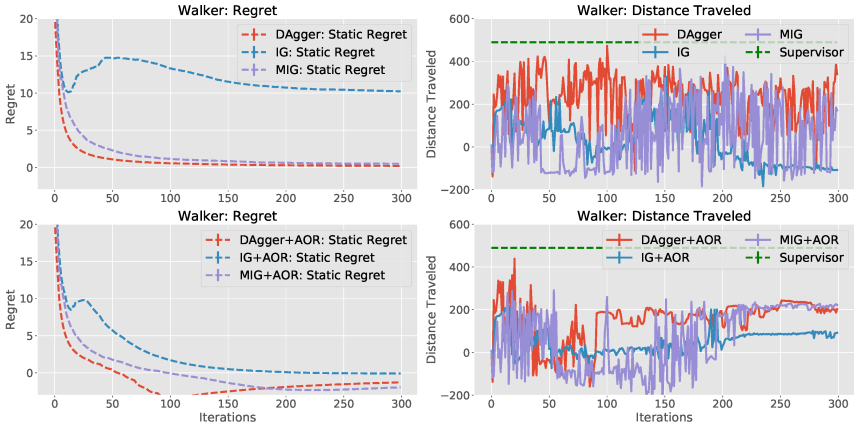

Fig. 9: The static regret is plotted against the number of iterations. As in the cart-pole environment, the regret descends to zero matter the agent's true performance on the system. The exception is the Imitation Gradient algorithm which achieves a seemingly high static regret without adaptive regularization. We hypothesize that proportionately large static regret is a result of suboptimal stepsizes for this particular algorithm in this domain. In theory, this regret converges over many iterations.

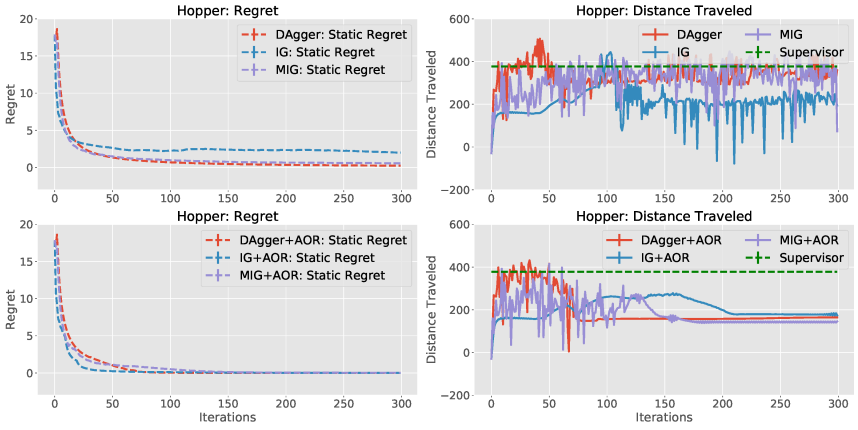

Fig. 10: For the Hopper environment, the static regret again tends to zero in all cases, showing it is not reflective of the true system costs.