Explaining Fast Improvement in Online Imitation Learning (Supplementary Material)

Xinyan Yan[1] **Byron Boots**[2] **Ching-An Cheng**[3]

[1]Aurora Innovation Inc., Pittsburgh, PA
[2]School of Computer Science and Engineering, University of Washington, Seattle, WA
[3]Microsoft Research, Redmond, WA

A PROOF OF TOOL LEMMAS

A.1 PROOF OF LEMMA 1

For completeness, we provide the proof for the basic inequality that upper bounds the norm of gradients by the function values, for smooth and nonnegative functions. This is essential for obtaining the self-bounding properties for proving Lemma 2 and Theorem 2 later on.

Lemma 1 (Lemma 3.1 [Srebro et al., 2010])**.** *Suppose a function $f : \mathcal{H} \to \mathbb{R}$ is β-smooth and non-negative, then for any $x \in \mathcal{H}$, $\|\nabla f(x)\|_*^2 \leq 4\beta f(x)$.*

Proof. Fix any $x \in \mathcal{H}$. And fix any $y \in \mathcal{H}$ satisfying $\|y - x\| \leq 1$. Let $g(u) = f(x + u(y - x))$ for any $u \in \mathbb{R}$. Fix any $u, v \in \mathbb{R}$,

$$\begin{aligned}|g'(v) - g'(u)| &= |\langle \nabla f(x + v(y - x)) - \nabla f(x + u(y - x)), y - x\rangle| \\ &\leq \|\nabla f(x + v(y - x)) - \nabla f(x + u(y - x))\|_* \|y - x\| \\ &\leq \beta |v - u| \|y - x\|^2 \\ &\leq \beta |v - u|\end{aligned}$$

Hence, g is β-smooth. By the mean-value theorem, for any $u, v \in \mathbb{R}$, there exists $w \in (u, v)$, such that $g(v) = g(u) + g'(w)(v - u)$. Hence

$$\begin{aligned}0 \leq g(v) &= g(u) + g'(u)(v - u) + (g'(w) - g'(u))(v - u) \\ &\leq g(u) + g'(u)(v - u) + \beta |w - u||v - u| \leq g(u) + g'(u)(v - u) + \beta (v - u)^2\end{aligned}$$

Setting $v = u - \frac{g'(u)}{2\beta}$ yields that $|g'(u)| \leq \sqrt{4\beta g(u)}$. Therefore, we have

$$|g'(0)| = |\langle \nabla f(x), y - x\rangle| \leq \sqrt{4\beta g(0)} = \sqrt{4\beta f(x)}$$

Therefore, by the definition of dual-norm,

$$\|\nabla f(x)\|_* = \sup_{y \in \mathcal{B}, \|y - x\| \leq 1} \langle \nabla f(x), y - x\rangle = \sup_{y \in \mathcal{B}, \|y - x\| \leq 1} |\langle \nabla f(x), y - x\rangle| \leq \sqrt{4\beta f(x)}$$

where the second equality is due to the domain of $y - x$. □

It's worthy to note that f needs to be smooth and non-negative on the entire Hilbert space $\mathcal{H}$.

B PROOF OF THEOREM 1

Theorem 1. *In Algorithm 1, suppose $\hat{l}_n$ is CSN and $\mathcal{A}$ is admissible. Let $\hat{\epsilon} = \frac{1}{N}\min_{\theta\in\Theta}\sum \hat{l}_n(\theta)$ be the bias, and let $\hat{E}$ be an upper bound on $\hat{\epsilon}$. Choose the stepsize η in $\mathcal{A}$ to be $\frac{1}{2\left(\beta+\sqrt{\beta^2+\frac{1}{2}\beta N\hat{E}R_{\mathcal{A}}^{-2}}\right)}$. Then it holds that*

$$\mathbb{E}\left[\frac{1}{N}\sum l_n(\theta_n) - \hat{\epsilon}\right] \leq \frac{8\beta R_{\mathcal{A}}^2}{N} + \sqrt{\frac{8\beta R_{\mathcal{A}}^2\hat{E}}{N}} \tag{8}$$

The rate (8) follows from analyzing the regret and the generalization error in the decomposition in (6). First, under the assumption of CSN loss functions and admissible online algorithms, the online regret can be bounded by an extension of the bias-dependent regret that is stated for mirror descent in [Srebro et al., 2010, Theorem 2], whose average gives the rate in (8) (see Appendix B.1). Second, the generalization error in (6) vanishes in expectation because it is a martingale difference sequence (see Appendix B.2).

B.1 UPPER BOUND OF ONLINE REGRET

We show a bias-dependent regret of admissible online algorithms (Definition 2) with CSN functions (Definition 3) by extending Theorem 2 of [Srebro et al., 2010] as follows.

Lemma 2. *Consider running an admissible online algorithm $\mathcal{A}$ on a sequence of CSN loss functions $\{f_n\}$. Let $\{\theta_n\}$ denote the online decisions made in each round, and let $\hat{\epsilon} = \frac{1}{N}\min_{\theta\in\Theta}\sum f_n(\theta)$ be the bias, and let $\hat{E}$ be such that $\hat{E} \geq \hat{\epsilon}$ almost surely. Choose η for $\mathcal{A}$ to be $\frac{1}{2\left(\beta+\sqrt{\beta^2+\frac{\beta N\hat{E}}{2R_{\mathcal{A}}^2}}\right)}$. Then the following holds*

$$\textit{Regret}(f_n) \leq 8\beta R_{\mathcal{A}}^2 + \sqrt{8\beta R_{\mathcal{A}}^2 N\hat{E}}.$$

Proof. Because the online algorithm $\mathcal{A}$ is admissible, we have

$$\text{Regret}(f_n) \leq \frac{1}{\eta}R_{\mathcal{A}}^2 + \frac{\eta}{2}\sum\|\nabla f_n(\theta_n)\|_*^2 \tag{11}$$

Let $\lambda = \frac{1}{2\eta}$ and $r^2 = 2R_{\mathcal{A}}^2$, then

$$\frac{1}{\eta}R_{\mathcal{A}}^2 + \frac{\eta}{2}\sum\|\nabla f_n(\theta_n)\|_*^2 = \lambda r^2 + \sum\frac{1}{4\lambda}\|\nabla f_n(\theta_n)\|_*^2 \tag{12}$$

Using Lemma 1 yields a *self-bounding property* for $\text{Regret}(f_n)$:

$$\text{Regret}(f_n) \leq \lambda r^2 + \frac{\beta}{\lambda}\sum f_n(\theta_n) \leq \lambda r^2 + \frac{\beta}{\lambda}\text{Regret}(f_n) + \frac{\beta}{\lambda}N\hat{E} \tag{13}$$

By rearranging the terms, we have a bias-dependent upper bound

$$\text{Regret}(f_n) \leq \frac{\beta}{\lambda-\beta}N\hat{E} + \frac{\lambda^2}{\lambda-\beta}r^2 \tag{14}$$

The upper bound can be minimized by choosing an optimal λ. Setting the derivative of the right-hand side to zero, and computing the optimal λ ($\lambda > 0$) gives us

$$r^2\lambda^2 - 2\beta r^2\lambda - \beta N\hat{E} = 0, \quad \lambda > 0 \quad \text{and} \quad \lambda = \beta + \sqrt{\beta^2 + \frac{\beta N\hat{E}}{r^2}} \tag{15}$$

which implies that the optimal η is $\frac{1}{2\left(\beta+\sqrt{\beta^2+\frac{\beta N\hat{E}}{2R_{\mathcal{A}}^2}}\right)}$. Since the optimal λ satisfies $\beta N\hat{E} = r^2\lambda^2 - 2\beta r^2\lambda$ implied from (15), (14) can be simplified into:

$$\begin{aligned}\text{Regret}(f_n) &\leq \frac{1}{\lambda-\beta}\beta N\hat{E} + \frac{\lambda^2}{\lambda-\beta}r^2 = \frac{1}{\lambda-\beta}(r^2\lambda^2 - 2\beta r^2\lambda) + \frac{\lambda^2}{\lambda-\beta}r^2 \\ &= \frac{2\lambda^2 r^2 - 2\beta\lambda r^2}{\lambda-\beta} = 2\lambda r^2\end{aligned} \tag{16}$$

Plugging in the optimal λ yields

$$\begin{aligned}\text{Regret}(f_n) \leq 2\lambda r^2 &= 2\left(\beta + \sqrt{\beta^2 + \frac{\beta N\hat{E}}{r^2}}\right) r^2 \\ &= 2\beta r^2 + \sqrt{2\beta r^2}\sqrt{2\beta r^2 + 2N\hat{E}} \\ &\leq 4\beta r^2 + 2\sqrt{\beta r^2 N\hat{E}} \\ &= 8\beta R_{\mathcal{A}}^2 + \sqrt{8\beta R_{\mathcal{A}}^2 N\hat{E}} \end{aligned} \tag{17}$$

where the last inequality uses the basic inequality: $\sqrt{a+b} \leq \sqrt{a} + \sqrt{b}$. □

Notably, the admissibility defined in Definition 2 is satisfied by common online algorithms, such as mirror descent [Nemirovski et al., 2009] and Follow-The-Regularized-Leader [McMahan, 2017] under first-order or full-information feedback, where η in Definition 2 corresponds to a constant stepsize, and $R_{\mathcal{A}}$ measures the size of the decision set Θ. More concretely, assume that the loss functions $\{f_n\}$ are convex. Then for mirror descent, with constant stepsize η, i.e., $\theta_{n+1} = \arg\min_{\theta\in\Theta} f_n(\theta) + \frac{1}{\eta} D_h(\theta||\theta_n)$, where h is 1-strongly convex and D_h is the Bregman distance generated by h defined by $D_h(x||y) = h(x) - h(y) - \langle \nabla h(y), x - y\rangle$ [Teboulle, 2018], $R_{\mathcal{A}}^2$ can be set to $\max_{x,y\in\Theta} D_h(x||y)$. And for FTRL with constant stepsize η, i.e., $\theta_{n+1} = \arg\min_{\theta\in\Theta} \sum f_n(\theta) + \frac{1}{\eta} h(\theta)$, where h is 1-strongly convex and non-negative, $R_{\mathcal{A}}^2$ can be set to $\max_{\theta\in\Theta} h(\theta)$ [McMahan, 2017, Theorem 1].

B.2 THE GENERALIZATION ERROR VANISHES IN EXPECTATION

The generalization error in (6) vanishes in expectation because it is a martingale difference sequence.

Lemma 3. *For Algorithm 1, the following holds:* $\mathbb{E}[\sum l_n(\theta_n) - \sum \hat{l}_n(\theta_n)] = 0$.

Proof. We show this by working from the end of the sequence. For brevity, we use the symbol colon in the subscript to represent a set that includes the start and the end indices, e.g. $\hat{l}_{1:N-2}$ stands for $\{\hat{l}_1, \ldots, \hat{l}_{N-2}\}$.

$$\begin{aligned}\mathbb{E}_{\hat{l}_{1:N}}\left[\sum_{t=1}^{N} l_n(\theta_n)\right] &= \mathbb{E}_{\hat{l}_{1:N-1}}\left[\sum_{t=1}^{N-1} l_n(\theta_n) + l_N(\theta_N)\right] \\ &= \mathbb{E}_{\hat{l}_{1:N-1}}\left[\sum_{t=1}^{N-1} l_n(\theta_n) + \mathbb{E}_{\hat{l}_N|\hat{l}_{1:N-1}}\left[\hat{l}_N(\theta_N)\right]\right] \\ &= \mathbb{E}_{\hat{l}_{1:N-2}}\left[\sum_{t=1}^{N-2} l_n(\theta_n) + l_{N-1}(\theta_{N-1}) + \mathbb{E}_{\hat{l}_{N-1:N}|\hat{l}_{1:N-2}}\left[\hat{l}_N(\theta_N)\right]\right] \\ &= \mathbb{E}_{\hat{l}_{1:N-2}}\left[\sum_{t=1}^{N-2} l_n(\theta_n) + \mathbb{E}_{\hat{l}_{N-1}|\hat{l}_{1:N-2}}\left[\hat{l}_{N-1}(\theta_{N-1})\right] + \mathbb{E}_{\hat{l}_{N-1:N}|\hat{l}_{1:N-2}}\left[\hat{l}_N(\theta_N)\right]\right] \\ &= \mathbb{E}_{\hat{l}_{1:N-2}}\left[\sum_{t=1}^{N-2} l_n(\theta_n) + \mathbb{E}_{\hat{l}_{N-1:N}|\hat{l}_{1:N-2}}\left[\sum_{t=N-1}^{N} \hat{l}_n(\theta_n)\right]\right]\end{aligned}$$

By applying the steps above repeatedly, the desired equality can be obtained. □

B.3 PUTTING TOGETHER

Finally, plugging Lemma 2 and Lemma 3 into (6) yields (8).

C PROOF OF THEOREM 2

Theorem 2. *Under the same assumptions and setup of Theorem 1, further assume that there is $G \in [0, \infty)$ such that, for any $\theta \in \Theta$, $\|\nabla \hat{l}_n(\theta)\|_* \leq G$. For any $\delta < 1/e$, with probability at least $1 - \delta$, the following holds*

$$\frac{1}{N}\sum l_n(\theta_n) - \epsilon = O\left(\frac{C\beta R^2}{N} + \sqrt{\frac{C\beta R^2(\hat{E}+\epsilon)}{N}}\right) \tag{9}$$

where $R_\Theta = \max_{\theta\in\Theta} \|\theta\|, R = \max(1, R_\Theta, R_\mathcal{A}), C = \log(1/\delta)\log(GRN)$.

C.1 DECOMPOSITION

The key to avoid the slow rate due to the direct application of martingale concentration analyses on the MDSs in (6) and (7) is to take a different decomposition of the cumulative loss. Here we construct two *new* MDSs in terms of the gradients: recall $\epsilon = \min_{\theta\in\Theta}\sum l_n(\theta)$ and let $\theta^\star = \arg\min_{\theta\in\Theta}\sum l_n(\theta)$. Then by convexity of l_n, we can derive

$$\begin{aligned}
&\sum l_n(\theta_n) - N\epsilon \\
\leq &\sum \langle \nabla l_n(\theta_n), \theta_n - \theta^\star\rangle \\
= &\sum \langle \nabla l_n(\theta_n) - \nabla \hat{l}_n(\theta_n), \theta_n - \theta^\star\rangle + \sum \langle \nabla \hat{l}_n(\theta_n), \theta_n - \theta^\star\rangle \\
\leq &\sum \underbrace{\langle \nabla l_n(\theta_n) - \nabla \hat{l}_n(\theta_n), \theta_n\rangle}_{\text{MDS}} - \sum \underbrace{\langle \nabla l_n(\theta_n) - \nabla \hat{l}_n(\theta_n)}_{\text{MDS}}, \theta^\star\rangle + \text{Regret}(\langle \nabla \hat{l}_n(\theta_n), \cdot\rangle)
\end{aligned} \tag{18}$$

Our proof is based on analyzing these three terms. The two MDSs are analyzed in Appendix C.2 and the regret is analyzed in Appendix C.3.

C.2 UPPER BOUND OF THE MARTINGALE CONCENTRATION

For the MDSs in (18), we notice that, for smooth and non-negative functions, the squared norm of the gradient can be bounded by the corresponding function value through Lemma 1. This enables us to properly control the second-order statistics of the MDSs in (18). By a recent vector-valued martingale concentration inequality that depends only on the second-order statistics [Rakhlin and Sridharan, 2015], we obtain a self-bounding property for (18) to get a fast concentration rate. The martingale concentration inequality is stated in the following lemma.

Lemma 4 (Theorem 3 [Rakhlin and Sridharan, 2015])**.** *Let $\mathcal{K}$ be a Hilbert space with norm $\|\cdot\|$ whose dual is $\|\cdot\|_*$. Let $\{z_t\}$ be a $\mathcal{K}$-valued martingale difference sequence with respect to $\{y_t\}$, i.e., $\mathbb{E}_{z_t|y_1,\ldots,y_{t-1}}[z_t] = 0$, and let h be a 1-strongly convex function with respect to norm $\|\cdot\|$ and let $B^2 = \sup_{x,y\in\mathcal{K},\|x\|=1,\|y\|=1} D_h(x\|y)$. Then for $\delta \leq 1/e$, with probability at least $1 - \delta$, the following holds*

$$\left\|\sum z_t\right\|_* \leq 2B\sqrt{V} + \sqrt{2\log(1/\delta)}\sqrt{1 + 1/2\log(2V + 2W + 1)}\sqrt{2V + 2W + 1}$$

where $V = \sum \|z_t\|_*^2$ and $W = \sum \mathbb{E}_{z_t|y_1,\ldots,y_{t-1}}\|z_t\|_*^2$.

In order to apply Lemma 4 to the MDSs in (18), the key is to properly upper bound the statistics V and W in Lemma 4 for these MDSs.

C.2.1 Upper Bound of the Concetration for MDS $\langle \nabla l_n(\theta_n) - \nabla \hat{l}_n(\theta_n), \theta_n\rangle$

Suppose that the decision set Θ is inside a ball centered at the origin in $\mathcal{H}$ with radius R_Θ.

Assumption 1. There exists $R_\Theta \in [0, \infty)$, such that $\max_{\theta\in\Theta}\|\theta\| \leq R_\Theta$.

Then by the definition of V and W in Lemma 4, and the definitions of the two problem-dependent policy class biases ϵ and $\hat{\epsilon}$ (see Definition 1), one can obtain

$$\begin{aligned}
V &= \sum |\langle \nabla l_n(\theta_n) - \nabla \hat{l}_n(\theta_n), \theta_n \rangle|^2 \\
&\leq \sum R_\Theta^2 \|\nabla l_n(\theta_n) - \nabla \hat{l}_n(\theta_n)\|_*^2 && \text{Assumption 1} \\
&\leq \sum R_\Theta^2 \left(2\|\nabla l_n(\theta_n)\|_*^2 + 2\|\nabla \hat{l}_n(\theta_n)\|_*^2\right) && \text{triangle inequality} \quad (19) \\
&\leq \sum R_\Theta^2 \left(8\beta l_n(\theta_n) + 8\beta \hat{l}_n(\theta_n)\right) && \text{Lemma 1} \\
&= 8\beta R_\Theta^2 (\text{Regret}(l_n) + \text{Regret}(\hat{l}_n) + N\epsilon + N\hat{\epsilon}) && \text{Definition 1} \quad (20)
\end{aligned}$$

Similarly for W, we have

$$\begin{aligned}
W &= \sum \mathbb{E}_{\hat{l}_n|\theta_n}[|\langle \nabla l_n(\theta_n) - \nabla \hat{l}_n(\theta_n), \theta_n \rangle|^2] \\
&\leq \sum R_\Theta^2 \mathbb{E}_{\hat{l}_n|\theta_n}[\|\nabla l_n(\theta_n) - \nabla \hat{l}_n(\theta_n)\|_*^2] && \text{Assumption 1} \\
&\leq \sum R_\Theta^2 \left(2\|\nabla l_n(\theta_n)\|_*^2 + 2\mathbb{E}_{\hat{l}_n|\theta_n}[\|\nabla \hat{l}_n(\theta_n)\|_*^2]\right) && \text{triangle inequality} \quad (21) \\
&\leq \sum R_\Theta^2 \left(8\beta l_n(\theta_n) + 8\mathbb{E}_{\hat{l}_n|\theta_n}[\beta \hat{l}_n(\theta_n)]\right) && \text{Lemma 1} \\
&= \sum R_\Theta^2 \left(8\beta l_n(\theta_n) + 8\beta l_n(\theta_n)\right) \\
&= 16\beta R_\Theta^2 (\text{Regret}(l_n) + N\epsilon) && \text{Definition 1} \quad (22)
\end{aligned}$$

Therefore,

$$V + W \leq 24\beta R_\Theta^2 (\text{Regret}(l_n) + \text{Regret}(\hat{l}_n) + N\epsilon + N\hat{\epsilon}) \tag{23}$$

Further suppose that the gradient of the sampled loss can be uniformly bounded:

Assumption 2. For any loss sequence $\{\hat{l}_n\}$ that can be experienced by Algorithm 1, suppose that there is $G \in [0, \infty)$ such that, for any $\theta \in \Theta$, $\|\nabla \hat{l}_n(\theta)\|_* \leq G$.

Then due to (23), $V \leq 4G^2 R_\Theta^2 N$ and $W \leq 4G^2 R_\Theta^2 N$. Now we are ready to invoke Lemma 4 by letting the Hilbert space $\mathcal{K}$ in Lemma 4 be $\mathbb{R}$, and denoting the corresponding B in Lemma 4 by $B_\mathbb{R}$. Then, for $\delta > 1/e$, with probability at least $1 - \delta$, the following holds

$$\begin{aligned}
&|\sum \langle \nabla l_n(\theta_n) - \nabla \hat{l}_n(\theta_n), \theta_n \rangle| \\
&\leq 2B_\mathbb{R} \sqrt{8\beta R_\Theta^2} \sqrt{\text{Regret}(l_n) + \text{Regret}(\hat{l}_n) + N\epsilon + N\hat{\epsilon}} + \\
&\sqrt{96\beta R_\Theta^2 \log(1/\delta)} \sqrt{1 + 1/2 \log(16G^2 R_\Theta^2 N + 1)} \sqrt{\text{Regret}(l_n) + N\epsilon + \text{Regret}(\hat{l}_n) + N\hat{\epsilon} + 1/(48\beta R_\Theta^2)}
\end{aligned} \tag{24}$$

C.2.2 Upper Bound of the Concetration for MDS $\nabla l_n(\theta_n) - \nabla \hat{l}_n(\theta_n)$

To bound $\|\sum \nabla l_n(\theta_n) - \nabla \hat{l}_n(\theta_n)\|_*$ that appears in

$$\sum \langle \nabla l_n(\theta_n) - \nabla \hat{l}_n(\theta_n), \theta^\star \rangle \leq R_\Theta \|\sum \nabla l_n(\theta_n) - \nabla \hat{l}_n(\theta_n)\|_* \tag{25}$$

We use Lemma 4 again in a similar way of deriving (24), except that this time the MDS $\nabla l_n(\theta_n) - \nabla \hat{l}_n(\theta_n)$ is vector-valued. Akin to showing (20) and (22), the statistics V can be bounded as

$$\begin{aligned}
V &\leq \sum \left(2\|\nabla l_n(\theta_n)\|_*^2 + 2\|\nabla \hat{l}_n(\theta_n)\|_*^2\right) && (26) \\
&\leq 8\beta (\text{Regret}(l_n) + \text{Regret}(\hat{l}_n) + N\epsilon + N\hat{\epsilon}) && (27)
\end{aligned}$$

and similarly for W:

$$W \leq \sum \left(2\|\nabla l_n(\theta_n)\|_*^2 + 2\mathbb{E}_{\hat{l}_n|\theta_n}[\|\nabla \hat{l}_n(\theta_n)\|_*^2]\right) \tag{28}$$

$$\leq 16\beta(\text{Regret}(l_n) + N\epsilon) \tag{29}$$

Therefore,

$$V + W \leq 24\beta(\text{Regret}(l_n) + \text{Regret}(\hat{l}_n) + N\epsilon + N\hat{\epsilon}) \tag{30}$$

Furthermore, by Assumption 2, it can be shown from (26) and (28) that $V \leq 4G^2N$ and $W \leq 4G^2N$. To invoke Lemma 4, let $\mathcal{K}$ in Lemma 4 be $\mathcal{H}$, and denote the corresponding B in Lemma 4 by $B_{\mathcal{H}}$. Then, for $\delta < 1/e$, with probability at least $1 - \delta$, the following holds

$$\begin{aligned}
&\|\sum \nabla l_n(\theta_n) - \nabla \hat{l}_n(\theta_n)\|_* \\
&\leq 2B_{\mathcal{H}}\sqrt{8\beta}\sqrt{\text{Regret}(l_n) + \text{Regret}(\hat{l}_n) + N\epsilon + N\hat{\epsilon}} + \\
&\sqrt{96\beta\log(1/\delta)}\sqrt{1 + 1/2\log(16G^2N+1)}\sqrt{\text{Regret}(l_n) + N\epsilon + \text{Regret}(\hat{l}_n) + N\hat{\epsilon} + 1/(48\beta)}
\end{aligned} \tag{31}$$

C.3 UPPER BOUND OF THE REGRET

Besides analyzing the MDSs, we need to bound the regret to the linear functions defined by the gradients (the last term in (18)). Since this last term is linear, not CSN, the bias-dependent online regret bound in the proof of Theorem 1 does not apply. Nonetheless, because these linear functions are based on the gradients of CSN functions, we discover that their regret rate actually obeys the exact same rate as the regret to the CSN loss functions. This is notable because the regret to these linear functions upper bounds the regret to the CSN loss functions.

Lemma 5. *Under the same assumptions and setup in Lemma 2,*

$$\textit{Regret}(\langle \nabla f_n(\theta_n), \cdot \rangle) \leq 8\beta R_{\mathcal{A}}^2 + \sqrt{8\beta R_{\mathcal{A}}^2 N\hat{E}}. \tag{32}$$

Proof. It suffices to show a self-bounding property for $\text{Regret}(\langle \nabla f_n(\theta_n), \cdot \rangle)$ as (13). Once this is established, the rest resembles how (17) follows from (13) through algebraic manipulations. As in Lemma 2, define $\lambda = \frac{1}{2\eta}$ and $r^2 = 2R_{\mathcal{A}}^2$. Due to the property of admissible online algorithms, one can obtain

$$\text{Regret}(\langle \nabla f_n(\theta_n), \cdot \rangle) \leq \frac{1}{\eta}R_{\mathcal{A}}^2 + \frac{\eta}{2}\sum \|\nabla f_n(\theta_n)\|_*^2 = \lambda r^2 + \sum \frac{1}{4\lambda}\|\nabla f_n(\theta_n)\|_*^2 \tag{33}$$

To proceed, as in Lemma 2, let $\hat{\epsilon} = \frac{1}{N}\min_{\theta\in\Theta}\sum \hat{l}_n(\theta)$ be the bias, and let $\hat{E}$ be such that $\hat{E} \geq \hat{\epsilon}$ almost surely. Using Lemma 1 and the admissibility of online algorithm $\mathcal{A}$ yields a self-bounding property for $\text{Regret}(\langle \nabla f_n(\theta_n), \cdot \rangle)$:

$$\begin{aligned}
\text{Regret}(\langle \nabla f_n(\theta_n), \cdot \rangle) &\leq \lambda r^2 + \frac{\beta}{\lambda}\sum f_n(\theta_n) \\
&\leq \lambda r^2 + \frac{\beta}{\lambda}\text{Regret}(f_n) + \frac{\beta}{\lambda}N\hat{E} \\
&\leq \lambda r^2 + \frac{\beta}{\lambda}\text{Regret}(\langle \nabla f_n(\theta_n), \cdot \rangle) + \frac{\beta}{\lambda}N\hat{E}
\end{aligned}$$

This self-bounding property is exactly like what we have seen in the self-bounding property for $\text{Regret}(f_n)$. After rearranging and computing the optimal λ (which coincides with the optimal λ in Lemma 2), (32) follows. □

Lemma 5 provides a bias-dependent regret to the linear functions defined by the gradients when the (stepsize) constant η is set optimally in the online algorithm $\mathcal{A}$ (used in Algorithm 1). Interestingly, the optimal η that achieves the bias-dependent regret coincides with the one for achieving a bias-dependent regret to CSN functions. Therefore, a bias-dependent bound for $\text{Regret}(\hat{l}_n)$ and $\text{Regret}(\langle \nabla \hat{l}_n(\theta_n), \cdot \rangle)$ can be achieved simultaneously.

C.4 PUTTING THINGS TOGETHER

We now have all the pieces to prove Theorem 2. Plugging (24), (25), and (31) into the decomposition (18), we have, for $\delta < 1/e$, with probability at least $1 - 2\delta$

$$
\begin{aligned}
&\text{Regret}(l_n) \\
&\leq 2B_{\mathbb{R}}\sqrt{8\beta R_\Theta^2}\sqrt{\text{Regret}(l_n) + \text{Regret}(\hat{l}_n) + N\epsilon + N\hat{\epsilon}} \\
&+ \sqrt{96\beta R_\Theta^2 \log(1/\delta)}\sqrt{1 + 1/2\log(16G^2R_\Theta^2 N + 1)}\sqrt{\text{Regret}(l_n) + N\epsilon + \text{Regret}(\hat{l}_n) + N\hat{\epsilon} + 1/(48\beta R_\Theta^2)} \\
&+ 2B_{\mathcal{H}}\sqrt{8\beta R_\Theta^2}\sqrt{\text{Regret}(l_n) + \text{Regret}(\hat{l}_n) + N\epsilon + N\hat{\epsilon}} \\
&+ \sqrt{96\beta \log(1/\delta)}\sqrt{1 + 1/2\log(16G^2 N + 1)}\sqrt{\text{Regret}(l_n) + N\epsilon + \text{Regret}(\hat{l}_n) + N\hat{\epsilon} + 1/(48\beta)} \\
&+ \text{Regret}(\langle \nabla \hat{l}_n(\theta_n), \cdot \rangle)
\end{aligned}
$$

To simplify it, we denote

$$
\begin{aligned}
A_1 &= 8\max(B_{\mathbb{R}}, B_{\mathcal{H}})\sqrt{2\beta R_\Theta^2}, \\
A_2 &= 8\sqrt{6\beta R_\Theta^2 \log(1/\delta)}\sqrt{1 + 1/2\log(16G^2\max(1, R_\Theta^2)N + 1)}, \\
\tilde{R} &= \min(1, R_\Theta)
\end{aligned}
$$

Plugging them into the above upper bound on $\text{Regret}(l_n)$ and using the basic inequality $\sqrt{a+b} \leq \sqrt{a} + \sqrt{b}$ yield

$$
\begin{aligned}
\text{Regret}(l_n) \leq (A_1 + A_2)\sqrt{\text{Regret}(l_n) + \text{Regret}(\hat{l}_n)} + (A_1 + A_2)\sqrt{N\epsilon + N\hat{\epsilon}} \\
+ \frac{A_2}{\sqrt{48\beta\tilde{R}^2}} + \text{Regret}(\langle \nabla \hat{l}_n(\theta_n), \cdot \rangle)
\end{aligned}
$$

To further simplify, using the basic inequality $\sqrt{ab} \leq (a+b)/2$ yields

$$
\begin{aligned}
\text{Regret}(l_n) \leq \frac{\text{Regret}(l_n)}{2} + \frac{\text{Regret}(\hat{l}_n)}{2} + (A_1 + A_2)\sqrt{N\epsilon + N\hat{\epsilon}} \\
+ \frac{(A_1 + A_2)^2}{2} + \frac{A_2}{\sqrt{48\beta\tilde{R}^2}} + \text{Regret}(\langle \nabla \hat{l}_n(\theta_n), \cdot \rangle)
\end{aligned}
$$

Rearranging terms and invoking the bias-dependent rate in Lemma 2 and Lemma 5 give

$$
\begin{aligned}
\text{Regret}(l_n) &\leq \text{Regret}(\hat{l}_n) + 2(A_1 + A_2)\sqrt{N\epsilon + N\hat{\epsilon}} + (A_1 + A_2)^2 + \frac{A_2}{\sqrt{12\beta\tilde{R}^2}} + 2\text{Regret}(\langle \nabla \hat{l}_n(\theta_n), \cdot \rangle) \\
&\leq 2(A_1 + A_2)\sqrt{N\epsilon + N\hat{\epsilon}} + 6\sqrt{2\beta R_{\mathcal{A}}^2 N\hat{E}} + (A_1 + A_2)^2 + \frac{A_2}{\sqrt{12\beta\tilde{R}^2}} + 24\beta R_{\mathcal{A}}^2
\end{aligned} \tag{34}
$$

Finally, to derive a big-O bound, denote

$$
R = \max(1, R_\Theta, R_{\mathcal{A}}), \quad C = \log(1/\delta)\log(GRN)
$$

then one can obtain the rate *in terms of* N in big-O notation, while keeping $\tilde{R}$, R, $B_{\mathbb{R}}$, $B_{\mathcal{H}}$, $\log(1/\delta)$, G, ϵ, and $\hat{E}$ as multipliers:

$$
\text{Regret}(l_n) = O\left(C\beta R^2 + \sqrt{C\beta R^2 N(\hat{E} + \epsilon)}\right)
$$

Therefore

$$
\frac{1}{N}\sum l_n(\theta_n) - \epsilon = O\left(\frac{C\beta R^2}{N} + \sqrt{\frac{C\beta R^2(\hat{E} + \epsilon)}{N}}\right)
$$

D ONLINE IL WITH ADAPTIVE STEPSIZES

In Appendix B and Appendix C, we proved new bias-dependent rates in expectation (Theorem 1) and in high-probability (Theorem 2). However, these rates hold only provided that the stepsize of the online algorithm $\mathcal{A}$ in Algorithm 1 is constant and properly tuned; this requires knowing in advance the smoothness factor β, an upper bound of the bias $\hat{\epsilon}$, and the number of rounds N. Therefore, these theorems are not directly applicable to practical online IL algorithms that update the stepsize adaptively without knowing the constants beforehand.

Fortunately, Theorem 1 and Theorem 2 can be adapted to online IL algorithms that utilize online algorithms with adaptive stepsizes in a straightforward manner, which we shall show next. The key insight is that online algorithms with adaptive stepsizes obtain almost the same guarantee as they would have known the optimal constant stepsize in advance. For example, Orabona [2019, Theorem 4.14] shows that for Online Subgradient Descent [Zinkevich, 2003], the difference between the guarantees of using the optimal constant stepsize and the guarantee of using adaptive stepsizes $\eta_n = \frac{\sqrt{2}D}{2\sqrt{\sum_{i=1}^n \|g_i\|_2^2}}$ is only a factor of $\sqrt{2}$.

Lemma 6 (Theorem 4.14 [Orabona, 2019])**.** *Let $V \subseteq \mathbb{R}^d$ a closed non-empty convex set with diameter D, i.e., $\max_{x,y\in V} \|x-y\|_2 \leq D$. Let $f_1, \dots, f_N$ be an arbitrary sequence of non-negative convex functions $f_n : \mathbb{R}^d \to (-\infty, +\infty]$ differentiable in open sets containing V for $t = 1, \dots, T$. Pick any $x_1 \in V$ and stepsize $\eta_n = \frac{\sqrt{2}D}{2\sqrt{\sum_{i=1}^n \|\nabla f_i(x_i)\|_2^2}}$, $n = 1, \dots, N$. Then the following regret bound holds for online subgradient descent:*

$$\text{Regret}(f_n) \leq \sqrt{2} \min_{\eta>0} \left(\frac{D^2}{2\eta} + \frac{\eta}{2} \sum_{n=1}^{T} \|\nabla f_n(x_n)\|_2^2 \right) \tag{35}$$

Inspired by this insight, we propose a more general definition of admissible online algorithms (cf. Definition 2) and a notion of proper stepsizes:

Definition 4 (General admissible online algorithm)**.** We say an online algorithm $\mathcal{A}$ is *admissible* on a parameter space Θ, if there exists $R_{\mathcal{A}} \in [0, \infty)$ such that given any sequence of differentiable convex functions f_n and stepsizes η_n, $\mathcal{A}$ can achieve $\text{Regret}(f_n) \leq \text{Regret}(\langle \nabla f_n(\theta_n), \cdot \rangle) \leq \frac{1}{\eta} R_{\mathcal{A}}^2 + \frac{1}{2} \sum \eta_n \|\nabla f_n(\theta_n)\|_*^2$, where θ_n is the decision made by $\mathcal{A}$ in round n.

The admissible online algorithms that we discussed in the last paragraph of Appendix B.1 also belong to this category of genearal admissible algorithms.

Definition 5 (Proper stepsizes)**.** A stepsize adaptation rule is *proper* if there exists $K \in (0, \infty)$ such that for any admissible online algorithm $\mathcal{A}$ (Definition 4) with the stepsize η_n chosen according to the rule based on the information till round n can achieve $\text{Regret}(f_n) \leq \text{Regret}(\langle \nabla f_n(\theta_n), \cdot \rangle) \leq K \min_{\eta>0} \left(\frac{1}{\eta} R_{\mathcal{A}}^2 + \frac{1}{2} \sum \eta \|\nabla f_n(\theta_n)\|_*^2 \right)$.

With the general definition of admissible online algorithms (Definition 4) and the definition of proper stepsizes (Definition 5), we can now extend the bias-dependent regret (Lemma 2) which assumes optimal constant stepsize to adaptive online algorithms with proper stepsizes. This lemma will be the foundation of extending Theorem 1 and Theorem 2.

Lemma 7. *Consider running an admissible online algorithm $\mathcal{A}$ on a sequence of CSN loss functions $\{f_n\}$ with adaptive stepsizes that are proper. Let $\{\theta_n\}$ denote the online decisions made in each round, and let $\hat{\epsilon} = \frac{1}{N} \min_{\theta\in\Theta} \sum f_n(\theta)$ be the bias, and let $\hat{E}$ be such that $\hat{E} \geq \hat{\epsilon}$ almost surely. Then the following holds*

$$\textit{Regret}(f_n) \leq 8K^2 \beta R_{\mathcal{A}}^2 + \sqrt{8K^2 \beta R_{\mathcal{A}}^2 N \hat{E}}.$$

Proof. Because the online algorithm $\mathcal{A}$ is admissible and the stepsizes are proper, we have, for *any* $\eta > 0$

$$\text{Regret}(f_n) \leq \frac{K}{\eta} R_{\mathcal{A}}^2 + \frac{K\eta}{2} \sum \|\nabla f_n(\theta_n)\|_*^2 \tag{36}$$

Let $\lambda = \frac{1}{2\eta}$ and $r^2 = 2R_{\mathcal{A}}^2$, then

$$\frac{K}{\eta} R_{\mathcal{A}}^2 + \frac{K\eta}{2} \sum \|\nabla f_n(\theta_n)\|_*^2 = K\lambda r^2 + \sum \frac{K}{4\lambda} \|\nabla f_n(\theta_n)\|_*^2 \tag{37}$$

Using Lemma 1 yields a *self-bounding property* for Regret(f_n):

$$\text{Regret}(f_n) \leq K\lambda r^2 + \frac{K\beta}{\lambda}\sum f_n(\theta_n) \leq K\lambda r^2 + \frac{K\beta}{\lambda}\text{Regret}(f_n) + \frac{K\beta}{\lambda}N\hat{E} \tag{38}$$

Let $\hat{\beta} = K\beta$ and $\hat{r}^2 = Kr^2$, and by rearranging the terms, we have a bias-dependent upper bound (cf. (14)) that for any $\eta > 0$,

$$\text{Regret}(f_n) \leq \frac{\hat{\beta}}{\lambda - \hat{\beta}}N\hat{E} + \frac{\lambda^2}{\lambda - \hat{\beta}}\hat{r}^2 \tag{39}$$

The upper bound can be minimized by choosing an optimal λ. Setting the derivative of the right-hand side to zero, and computing the optimal λ ($\lambda > 0$) gives us

$$\hat{r}^2\lambda^2 - 2\hat{\beta}\hat{r}^2\lambda - \hat{\beta}N\hat{E} = 0, \quad \lambda > 0 \quad \text{and} \quad \lambda = \hat{\beta} + \sqrt{\hat{\beta}^2 + \frac{\hat{\beta}N\hat{E}}{\hat{r}^2}} \tag{40}$$

which implies that the optimal η is $\frac{1}{2\left(\hat{\beta}+\sqrt{\hat{\beta}^2+\frac{\hat{\beta}N\hat{E}}{2R_{\mathcal{A}}^2}}\right)}$. Because (39) holds for any η, it holds for the optimal η too. Next, we simplify (39). Since the optimal λ satisfies the equality $\hat{\beta}N\hat{E} = \hat{r}^2\lambda^2 - 2\hat{\beta}\hat{r}^2\lambda$ implied from (40), (39) can be written as

$$\begin{aligned}\text{Regret}(f_n) &\leq \frac{1}{\lambda - \hat{\beta}}\hat{\beta}N\hat{E} + \frac{\lambda^2}{\lambda - \hat{\beta}}\hat{r}^2 = \frac{1}{\lambda - \hat{\beta}}(\hat{r}^2\lambda^2 - 2\hat{\beta}\hat{r}^2\lambda) + \frac{\lambda^2}{\lambda - \hat{\beta}}\hat{r}^2 \\ &= \frac{2\lambda^2\hat{r}^2 - 2\hat{\beta}\lambda\hat{r}^2}{\lambda - \hat{\beta}} = 2\lambda\hat{r}^2\end{aligned} \tag{41}$$

Plugging in the optimal λ yields

$$\begin{aligned}\text{Regret}(f_n) &\leq 2\lambda\hat{r}^2 = 2\left(\hat{\beta} + \sqrt{\hat{\beta}^2 + \frac{\hat{\beta}N\hat{E}}{\hat{r}^2}}\right)\hat{r}^2 \\ &= 2\hat{\beta}\hat{r}^2 + \sqrt{2\hat{\beta}\hat{r}^2}\sqrt{2\hat{\beta}\hat{r}^2 + 2N\hat{E}} \\ &\leq 4\hat{\beta}\hat{r}^2 + 2\sqrt{\hat{\beta}\hat{r}^2N\hat{E}} \\ &= 8K^2\beta R_{\mathcal{A}}^2 + \sqrt{8K^2\beta R_{\mathcal{A}}^2 N\hat{E}}\end{aligned} \tag{42}$$

where the last inequality uses the basic inequality: $\sqrt{a+b} \leq \sqrt{a} + \sqrt{b}$. □

Provided the bias-dependent regret Lemma 7 (cf. Lemma 2), a bias-dependent rate in expectation for online IL with an additional constant K due to the adaptive stepsizes (cf. Theorem 1) follows directly, because Lemma 3 still holds even if the stepsizes of the online algorithm in Algorithm 1 become adaptive. In order to extend Theorem 2 to admissible online algorithm with adaptive stepsizes, we first need to derive a bias-dependent regret to the linear functions defined by the gradients (cf. Lemma 5) based on the proof of Lemma 7.

Lemma 8. *Under the same assumptions and setup in Lemma 7,*

$$\textit{Regret}(\langle \nabla f_n(\theta_n), \cdot \rangle) \leq 8K^2\beta R_{\mathcal{A}}^2 + \sqrt{8K^2\beta R_{\mathcal{A}}^2 N\hat{E}}. \tag{43}$$

Proof. It suffices to show a self-bounding property for Regret$(\langle \nabla f_n(\theta_n), \cdot \rangle)$ as (38). Once this is established, the rest resembles how (42) follows from (38) through algebraic manipulations. As in Lemma 7, define $\lambda = \frac{1}{2\eta}$ and $r^2 = 2R_{\mathcal{A}}^2$. Due to the property of admissible online algorithms, one can obtain, for any η

$$\text{Regret}(\langle \nabla f_n(\theta_n), \cdot \rangle) \leq \frac{K}{\eta}R_{\mathcal{A}}^2 + \frac{K\eta}{2}\sum \|\nabla f_n(\theta_n)\|_*^2 = K\lambda r^2 + \sum \frac{K}{4\lambda}\|\nabla f_n(\theta_n)\|_*^2 \tag{44}$$

To proceed, as in Lemma 2, let $\hat{\epsilon} = \frac{1}{N}\min_{\theta\in\Theta}\sum \hat{l}_n(\theta)$ be the bias, and let $\hat{E}$ be such that $\hat{E} \geq \hat{\epsilon}$ almost surely. Using Lemma 1 and the admissibility of online algorithm $\mathcal{A}$ yields a self-bounding property for Regret$(\langle \nabla f_n(\theta_n), \cdot\rangle)$:

$$\begin{aligned}
\text{Regret}(\langle \nabla f_n(\theta_n), \cdot\rangle) &\leq K\lambda r^2 + \frac{K\beta}{\lambda}\sum f_n(\theta_n) \\
&\leq K\lambda r^2 + \frac{K\beta}{\lambda}\text{Regret}(f_n) + \frac{K\beta}{\lambda}N\hat{E} \\
&\leq K\lambda r^2 + \frac{K\beta}{\lambda}\text{Regret}(\langle \nabla f_n(\theta_n), \cdot\rangle) + \frac{K\beta}{\lambda}N\hat{E}
\end{aligned}$$

This self-bounding property is exactly like what we have seen in the self-bounding property for Regret(f_n). After rearranging and computing the optimal λ (which coincides with the optimal λ in Lemma 2), (43) follows. □

Given the bias-dependent rates in online learning (Lemma 7 and Lemma 8), a high-probability bias-dependent rate for online IL with an additional constant K due to adaptive stepsizes (cf. Theorem 2) can the derived in the same way as the proof of Theorem 2, except that Lemma 7 and Lemma 8 will be invoked in (34) in place of Lemma 2 and Lemma 5.

Interestingly, Orabona [2019, Theorem 4.21] provides a bias-dependent regret for Online Subgradient Descent with adaptive stepsizes $\eta_n = \frac{\sqrt{2}D}{2\sqrt{\sum_{i=1}^n \|g_i\|_2^2}}$ (cf. Lemma 7). Although that regret bound would help more directly prove Theorem 1 in the adaptive stepsize setting, but it does not directly imply a bias-dependent regret to the linear functions defined by the gradients (cf. Lemma 8).

E EXPERIMENT DETAILS

Although the main focus of this paper is the new theoretical insights, we conduct experiments to provide evidence that the fast policy improvement phenomena indeed exist, as our theory predicts. We verify the change of the policy improvement rate due to policy class capacity by running an online IL experiment in the CartPole balancing task in OpenAI Gym [Brockman et al., 2016] with DART physics engine [Lee et al., 2018]. Details of reproducing the experimental results can be found in the README file in the supplementary materials.

E.1 MDP SETUP

The goal of the CartPole balancing task is to keep the pole upright by controlling the acceleration of the cart. This MDP has a 4-dimensional continuous state space (the position and the velocity of the cart and the pole), and 1-dimensional continuous action space (the acceleration of the cart). The initial state is a configuration with a small uniformly sampled offset from being static and vertical, and the dynamics is deterministic. This task has a maximum horizon of 1000. In each time step, if the pole is maintained within a threshold from being upright, the learner receives an instantaneous reward of one; otherwise, the learner receives zero reward and the episode terminates. Therefore, the maximum sum of rewards for an episode is 1000.

E.2 EXPERT POLICY REPRESENTATION AND TRAINING

To simulate the online IL task, we consider a neural network expert policy (with one hidden layer of 64 units and $\tanh$ activation), and the inputs to the neural network is normalized using a moving average over the samples. The expert policy is trained using a model-free policy gradient method (ADAM [Kingma and Ba, 2014] with GAE [Schulman et al., 2015]). And the value function used by GAE is represented by a neural network with two hidden layers of 128 units and $\tanh$ activation. To compute the policy gradient during training, additional Gaussian noise (with zero mean and a learnable variance that does not depend on the state) is added to the actions, and the gradient is computed through log likelihood ratio. After 100 rounds of training, the expert policy can consistently achieve the maximum sum of rewards both with and without the additional Gaussian noise. After the expert policy is trained, during online IL, Gaussian noise is not added in order to reduce the variance in the experiments.

E.3 LEARNER POLICY REPRESENTATION

We let the learner policy be another neural network that has exactly the same architecture as the expert policy with no Gaussian noise added. In the setting of only training the output layer, we copy the weights for the hidden layer and the input

normalizer from those of the expert policy and randomly initialize the weights of the output layer. During training, only the weights of the learner's output layer were updated. In this way, we can view the learner as a *linear* policy using the representation of the expert policy. In the setting of training the full network, we still copy the input normalizer from that of the expert policy but we randomly initialize all the variables in the network, i.e., weights and biases of the hidden and output layers. During training, all of these variables were updated.

E.4 ONLINE IL SETUP

Policy class We conduct online IL with unbiased and biased policy classes. One one hand, we define the unbiased class as all the policies satisfying the representation in Appendix E.3. On the other hand, we define the biased policy classes by imposing an additional ℓ_2-norm constraint with different sizes on the learner's weights in the output layer so that the learner cannot perfectly mimic the expert policy. More concretely, in the experiments, the ℓ_2-norm constraint has sizes $\{0.1, 0.12, 0.15\}$. This set of constraints was chosen based on the observation that the ℓ_2-norm of the final policy trained without the constraint is about 0.18 when training the output layer only and about 0.23 when training the full network.

Loss functions We select $l_n(\theta) = \mathbb{E}_{s\sim d_{\pi_{\theta_n}}}[H_\mu(\pi_\theta(s) - \pi_{\mathrm{e}}(s))]$ as the online IL loss (see Section 2.2), where H_μ is the Huber function defined as $H_\mu(x) = \frac{1}{2}x^2$ for $|x| \leq \mu$ and $\mu|x| - \frac{1}{2}\mu^2$ for $|x| > \mu$. In the experiments, μ is set to 0.05; as a result, H_μ is linear when its function value is larger than 0.00125. Because the learner's policy is linear, this online loss is CSN in the unknown weights of the learner.

Policy update rule We choose AdaGrad [McMahan and Streeter, 2010, Duchi et al., 2011] as the online algorithm in Algorithm 1; AdaGrad is a first-order mirror descent algorithm and well matches the assumptions made in our theorems (Appendix D), When the ℓ_2-norm constraint is imposed, an additional projection step is taken after taking a gradient step using AdaGrad. The final algorithm is a special case of the DAgger algorithm [Ross et al., 2011] (called DAggereD in [Cheng et al., 2018]) with only first-order information and continuous actions [Cheng et al., 2018]. In the experiments, the stepsize is set to 0.01. In each round, for updating the learner policy, 1000 samples, i.e., state and expert action pairs, are gathered, and for computing the loss $l_n(\theta_n)$, more samples (5000 samples) are used due to the randomness in the initial state of the MDP. The total number of iterations is 500 for both the training output layer and the training full network experiments. Due to the randomness in the initial state of the MDP and the initialization of the policy, we averaged the results over 4 random seeds.

Hyperparameter tuning The hyperparameters are tuned in a very coarse manner. We eliminated the ones that are obviously not proper. Here are the hyperparameters we have tried. The stepsize in online IL: $0.1, 0.01, 0.001$. The Huber function parameter μ: 0.05.

E.5 OTHER DETAILS

Computing infrastructure All the experiments were conducted on a desktop with Intel(R) Core(TM) i7-4770 CPU @ 3.40GHz, 32GB memory, and no GPU. The operating system is Ubuntu 16.04.

Average runtime On the aforementioned desktop, it took 15 min to train the expert, 45 min to do online IL.

References

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

Ching-An Cheng, Xinyan Yan, Nolan Wagener, and Byron Boots. Fast policy learning through imitation and reinforcement. In *Proceedings of the 34th Conference on Uncertanty in Artificial Intelligence*, pages 845–855, 2018.

John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of machine learning research*, 12(7), 2011.

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

Jeongseok Lee, Michael X. Grey, Sehoon Ha, Tobias Kunz, Sumit Jain, Yuting Ye, Siddhartha S. Srinivasa, Mike Stilman, and C. Karen Liu. DART: Dynamic animation and robotics toolkit. *The Journal of Open Source Software*, 3(22):500, feb 2018.

H Brendan McMahan. A survey of algorithms and analysis for adaptive online learning. *The Journal of Machine Learning Research*, 18(1):3117–3166, 2017.

H Brendan McMahan and Matthew Streeter. Adaptive bound optimization for online convex optimization. *arXiv preprint arXiv:1002.4908*, 2010.

Arkadi Nemirovski, Anatoli Juditsky, Guanghui Lan, and Alexander Shapiro. Robust stochastic approximation approach to stochastic programming. *SIAM Journal on optimization*, 19(4):1574–1609, 2009.

Francesco Orabona. A modern introduction to online learning. *arXiv preprint arXiv:1912.13213*, 2019.

Alexander Rakhlin and Karthik Sridharan. On equivalence of martingale tail bounds and deterministic regret inequalities. *arXiv preprint arXiv:1510.03925*, 2015.

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

John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation. *arXiv preprint arXiv:1506.02438*, 2015.

Nathan Srebro, Karthik Sridharan, and Ambuj Tewari. Smoothness, low noise and fast rates. In *Advances in neural information processing systems*, pages 2199–2207, 2010.

Marc Teboulle. A simplified view of first order methods for optimization. *Mathematical Programming*, 170(1):67–96, 2018.

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.