Optimism & Adaptation in Policy Optimization

Anonymous Author(s)
Affiliation
Address
email

Abstract

We work towards a unifying paradigm for accelerating policy optimization methods in reinforcement learning (RL) through *optimism* & *adaptation*. Leveraging the deep connection between policy iteration and policy gradient methods, we recast seemingly unrelated policy optimization algorithms as the repeated application of two interleaving steps (i) an *optimistic policy improvement operator* maps a prior policy π_t to a hypothesis π_{t+1} using a *gradient ascent prediction*, followed by (ii) a *hindsight adaptation* of the optimistic prediction based on a partial evaluation of the performance of π_{t+1}. We use this shared lens to jointly express other well-known algorithms, including soft and optimistic policy iteration, natural actor-critic methods, model-based policy improvement based on forward search, and meta-learning algorithms. By doing so, we shed light on collective theoretical properties related to acceleration via optimism & adaptation. Building on these insights, we design an *adaptive & optimistic policy gradient* algorithm via meta-gradient learning, and empirically highlight several design choices pertaining to optimism, in an illustrative task.

1 Introduction

Policy gradient (PG) methods [Williams, 1992, Sutton et al., 1999] are one of the most effective reinforcement learning (RL) algorithms [Espeholt et al., 2018, Schulman et al., 2015, 2017, Abdolmaleki et al., 2018, Hessel et al., 2021, Zahavy et al., 2020, Flennerhag et al., 2021]. These methods search for the optimal policy in a parametrized class of policies by using gradient ascent to maximize the cumulative expected reward that a policy collects when interacting with an environment. While effective, this objective poses challenges to the analysis and understanding of PG-based optimization algorithms due to its non-concavity in the policy parametrization [Agarwal et al., 2019, Mei et al., 2020b,a, 2021b].

Nevertheless, PG methods globally converge sub-linearly for smoothly parametrized softmax policy classes. This analysis relies on local linearization of the parameter vector and uses small step sizes and gradient domination to control the errors introduced from the linearization [Agarwal et al., 2019, Mei et al., 2020b, 2021b, 2023]. In contrast, policy iteration (PI) linearizes the objective w.r.t. (with respect to) the functional representation of the policy [Agarwal et al., 2019, Bhandari and Russo, 2019, 2021], and converges linearly. Relying on the illuminating connections between PI and several instances of PG algorithms (including natural policy gradients (NPG) and (inexact) mirror ascent (MA)), recent works [Bhandari and Russo, 2021, Cen et al., 2022, Mei et al., 2021a, Yuan et al., 2023] extended the above results and showed linear convergence of PG algorithms with large step sizes (adaptive or geometrically increasing). Other works showed that PG methods can achieve linear rates via entropy regularization. These guarantees cover some (approximately) closed policy classes, e.g., tabular, or log-linear—cf. Table 2 in Appendix A.

This recent understanding about the convergence properties of policy gradient methods in RL leaves room to consider more advanced techniques. In this work, we focus on **optimism**—a term we borrow from online convex optimization [Zinkevich, 2003], and is unrelated to the exploration strategy of

Submitted to 16th European Workshop on Reinforcement Learning (EWRL 2023). Do not distribute.

optimism in the face of uncertainty. In this context, *optimism* refers to predicting future gradients in order to accelerate convergence (for instance, as done in Nesterov's accelerated gradients (NAG) [Nesterov, 1983, Wang and Abernethy, 2018, Wang et al., 2021], extra-gradient (EG) methods [Korpelevich, 1976], mirror-prox [Nemirovski, 2004, Juditsky et al., 2011], optimistic MD [Rakhlin and Sridharan, 2013a, Joulani et al., 2020], AO-FTRL [Rakhlin and Sridharan, 2014, Mohri and Yang, 2015], etc.).

In RL, optimistic policy iteration (OPI) [Bertsekas and Tsitsiklis, 1996, Bertsekas, 2011, Tsitsiklis, 2002] considers policy updates performed based on incomplete evaluation, with a value function estimate that gradually tracks the solution of the most recent policy evaluation problem. Non-optimistic methods, on the other hand, regard the value estimation problem as a series of isolated *prediction* problems and solve them by Monte Carlo or temporal difference (TD) estimation. By doing so, they ignore the potentially predictable nature of the prediction problems, and their solutions, along a policy's optimization path.

In previous work, optimism has been studied in policy optimization to mitigate oscillations [Wagner, 2014, 2013, Moskovitz et al., 2023] as well as for accelerated optimization [Cheng et al., 2018, Hao et al., 2020], resulting in sub-linear convergence, cf. Table 2 in Appendix A.

In this paper, we introduce a framework for optimism and adaptation in policy optimization that allows us to describe seemingly disparate algorithms as making specific choices in how they represent, or adapt optimistic gradient predictions. In particular, we show that two classes of well-known algorithms—meta-learning algorithms and model-based planning algorithms—can be viewed as optimistic variants of vanilla policy optimization, and provide a theoretical argument for their empirical success. For example, STACX [Zahavy et al., 2020] represents an optimistic variant of Impala [Espeholt et al., 2018] and achieves a doubling of Impala's performance on the Atari-57 suite; similarly, adding further optimistic steps in BMG [Flennerhag et al., 2021] yields another doubling of the performance relative to that of STACX. In model-based RL, algorithms with extra steps of planning, e.g., the AlphaZero family of algorithms [Silver et al., 2016a, 2017], with perfect simulators, also enjoy huge success in challenging domains, e.g. chess, Go, and MuZero [Schrittwieser et al., 2019], with an *adaptive* model, achieves superhuman performance in challenging and visually complex domains.

Scope, purpose & outline After some background in Sec. 2, we present *optimism & adaptation* as a framework for analyzing policy optimization algorithms. This formulation resolves around two interleaving operators. Each operator executes a proximal (soft) update toward the most recent solution obtained by bootstrapping on the opposite operator

(i) *optimistic improvement* softly updates a prior policy π_t to a *proposed hypothesis* π_{t+1} in the direction given by a *prediction* of the gradient ascent direction—denoted with Q_t,

(ii) *hindsight adaptation* softly updates the previous *gradient prediction* Q_t toward the most recent *incomplete evaluation* of the proposed hypothesis π_{t+1}.

We show this two-step—*optimism & adaptation*—operator-view is a generalization of (i) soft policy iteration at one end, and (ii) optimistic policy iteration at the other. Soft policy iteration performs proximal updates for *policy improvement*, averaging the solutions from past updates, and for *adaptation* it simply evaluates the most recent solution π_{t+1}, without gradually averaging past solutions. Optimistic policy iteration does the reverse, *policy improvement* with hard greedification, and no averaging of past solutions, and *hindsight adaptation* with gradual averaging.

We use this formulation to highlight the commonalities among several algorithms, expressed in this formalism in Sec. 3, including inexact natural policy gradients, model-based policy optimization algorithms relying on run-time forward search (e.g. Silver et al. [2016a, 2017], Schrittwieser et al. [2019], Hessel et al. [2021]), and a general algorithm for *adaptive optimistic policy gradients* via *meta-gradient optimization* (common across the algorithmic implementations of Zahavy et al. [2020], Flennerhag et al. [2021]). We believe this unifying paradigm could be used in the design of other adaptive optimistic policy optimization algorithms, or guide the investigation into other collective properties of these methods.

From the analysis in Sec. 4, it transpires that modifying policy iteration to include online functional averaging—via proximal updates, and optimistic improvements—via predictions, has linear convergence, and the composition of proximal adaptation and improvement operators has properties relevant for accelerated learning. Leveraging these insights, in Sec. 5, we relax the assumptions of

policy iteration, and introduce a particular instance of an optimistic policy gradient algorithm that achieves adaptation via meta-gradient learning. In Sec. 6, we use an illustrative task to test several theoretical predictions empirically. First, we tease apart the role of optimism in forward search algorithms. Second, we analyze the properties of the adaptive optimistic algorithm we introduced in Sec. 5. We end with a discussion on the use of optimism & adaptation in a broader context, and the many questions left unanswered, in Sec. 7.

2 Background & notation

Setting We consider a standard reinforcement learning (RL) setting described by means of an infinite-horizon discounted Markov decision process (MDP) [Puterman, 1994] $\mathcal{M} \equiv \{\mathcal{S}, \mathcal{A}, r, P, \gamma, \rho\}$, with state space $\mathcal{S}$ and action space $\mathcal{A}$, discount factor $\gamma \in [0, 1)$, with initial states sampled under the initial distribution ρ. We assume throughout that $\rho(s) > 0 \forall sin \mathcal{S}$ Assuming an exploratory initial distribution is critical as it is well known that, in the absence of strong assumptions on the transition kernel, policy gradient methods can fail catastrophically if applied without some form of intelligent exploration. The agent follows an online learning protocol: at timestep $t \geq 0$, the agent is in state $S_t \in \mathcal{S}$, takes action $A_t \in \mathcal{A}$, given a policy $\pi_t(\cdot|s_t)$—the distribution over actions for each state $\pi : \mathcal{S} \to \Delta_{\mathcal{A}}$, with $\Delta_{\mathcal{A}}$—the action simplex—the space of probability distributions defined on $\mathcal{A}$. It then receives reward $R_t \sim r$ and transitions to a next state $S_{t+1} \sim P(\cdot|S_t, A_t)$. Let $d_\pi(s)$ be a measure over states, representing the discounted visitation distribution (or discounted fraction of time the system spends in a state s) $d_\pi(s) = (1-\gamma) \sum_{t=0}^{\infty} \gamma^t \Pr(S_t = s | S_0 \sim \rho, A_k \sim \pi(\cdot|S_k), \forall k \leq t)$, with $\Pr(S_t = s | S_0 \ldots)$ the probability of transitioning to a state at timestep t given policy π.

Problem formulation The RL problem consists in finding a policy π maximizing the discounted return—the policy performance objective: $V_\rho(\pi) \equiv \mathbb{E}_{S \sim \rho}[V_\pi(S)] = (1-\gamma)\mathbb{E}_{\pi,\rho}\big[\sum_{t \geq 0} \gamma^t R_{t+1}\big]$, where $V_\pi \in \mathbb{R}^{|\mathcal{S}|}$ is the value function, and $Q_\pi \in \mathbb{R}^{|\mathcal{S}| \times |\mathcal{A}|}$ the action-value function of a policy $\pi \in \Pi = \{\pi \in \mathbb{R}_+^{|\mathcal{S}| \times |\mathcal{A}|} | \sum_{a \in \mathcal{A}} \pi(s, a) = 1, \forall s \in \mathcal{S}\}$, s.t. $Q_\pi(s, a) \equiv \mathbb{E}_\pi \left[\sum_{t=0}^{\infty} \gamma^t R_t | S_0 = s, A_0 = a\right]$, and $V_\pi(s) \equiv \mathbb{E}_\pi \left[Q(s, A)\right]$. Let $\mathcal{T}_\pi : \mathbb{R}^{|\mathcal{S}|} \to \mathbb{R}^{|\mathcal{S}|}$ be the Bellman evaluation operator, and $\mathcal{T} : \mathbb{R}^{|\mathcal{S}|} \to \mathbb{R}^{|\mathcal{S}|}$ the Bellman optimality operator, s.t. $(\mathcal{T}_\pi V)(s) \equiv r(s, \pi(s)) + \gamma \sum_{s' \in \mathcal{S}} P(s'|s, \pi(s)) V(s')$, and $(\mathcal{T} V)(s) = \max_{a \in \mathcal{A}} r(s, a) + \gamma \sum_{s' \in \mathcal{S}} P(s'|s, a) V(s') = \max_{\pi \in \Pi} (\mathcal{T}_\pi V)(s)$, with Q-fn. analogs.

Algorithms The classic **policy iteration** algorithm repeats consecutive stages of (i) one-step greedy policy improvement w.r.t. a value function estimate $\pi_{t+1} \in \mathcal{G}(V_{\pi_t}) = \{\pi : \mathcal{T}_\pi V_{\pi_t} = \mathcal{T} V_{\pi_t}\}$, with $\mathcal{G}$ the greedy set of V_{π_t}, followed by (ii) evaluation of the value function w.r.t. the greedy policy $V_{\pi_{t+1}} = \lim_{h \to \infty} \mathcal{T}_{\pi_{t+1}}^h V_{\pi_t}$. Approximations of either steps lead to approximate PI (API) [Scherrer et al., 2015]. Relaxing the greedification leads to **soft PI** [Kakade and Langford, 2002]. $\pi_{t+1} = (1-\kappa)\pi_t + \kappa \pi_{t+1}^+$, with $\pi_{t+1}^+ = \arg\max_{\pi \in \Pi} \langle Q_{\pi_t}, \pi \rangle_{d_{\pi_t}}$ for $\kappa \in [0, 1]$, a (possibly time-dependent) step size, and $\langle \cdot, \cdot \rangle_d$ a state weighting that places weight $d(s)$ on any state-action pair (s, a).

Optimistic PI (OPI) [Bertsekas and Tsitsiklis, 1996] relaxes the evaluation step instead to $Q_{t+1} = (1-\lambda) Q_t + \lambda Q_{t+1}^+$, with $Q_{t+1}^+ = \mathcal{T} Q_t$. Using $Q_{t+1}^+ = \mathcal{T}^h Q_t, \forall h \geq 0$, and $Q_{t+1}^+ = \mathcal{T}^\lambda Q_t$, has been called modified PI [Puterman, 1994], and λ-PI [Scherrer and Thiery, 2010], respectively.

More commonly used in practice are **policy gradient** algorithms, and specifically **actor-critics**, which extend the search in PI to parameterized policy classes with softmax transforms $\Pi_\Theta = \big\{\pi_\theta \big| \pi_\theta(s, a) = \frac{\exp f_\theta(s,a)}{\sum_{a' \in \mathcal{A}} \exp f_\theta(s,a')} \forall s \in \mathcal{S}, a \in \mathcal{A}, \theta \in \mathbb{R}^m\big\}$, with f_θ a differentiable function, either tabular $f_\theta(s, a) = \theta_{s,a}$, log-linear $f_\theta(s, a) = \phi(s, a)^\top \theta$, with ϕ a feature representation, or neural (f_θ-a neural network) parametrizations [Agarwal et al., 2019]. These methods search over the parameter vector θ of a softmax policy $\pi_\theta \in \Pi_\Theta$, using as surrogate objectives, local linearizations of the performance, expressed as regularized iterative optimization $\theta_{t+1} = \arg\max_{\theta \in \Theta} \langle \pi, Q_t \rangle_{d_{\pi_t}} - 1/\alpha \, \mathrm{KL}_{[d_{\pi_t}]}(\pi, \pi_t)$ with $\mathrm{KL}_{[d]}(\cdot, \cdot)$ denoting a state weighting that places weight $d(s)$ on any state-action pair (s, a), and $\alpha \geq 0$ a step size, $\pi_t = \pi_{\theta_t} \in \Pi_\Theta$ a parametric policy class (tabular, log-linear or neural cf. [Agarwal et al., 2019]), and $Q_t \approx Q_{\pi_t}$, generally an estimated critic.

Model-based policy improvement algorithms based on forward search rely on approximate online versions of multi-step greedy improvement implemented via Monte Carlo Tree Search (MCTS) [Browne et al., 2012]. These algorithms replace the one-step greedy policy in the improvement stage of PI, or soft PI, with an h-greedy policy—a policy that is greedy w.r.t. a horizon of h steps, namely $\pi \in \mathcal{G}(\mathcal{T}^{h-1} V) = \{\pi | \mathcal{T}_\pi(\mathcal{T}^{h-1} V) = \mathcal{T}^h V\}$, or a κ-greedy policy $\pi \in \mathcal{G}(\mathcal{T}^\kappa V) = \{\pi | \mathcal{T}_\pi^\kappa(V) =$

$\mathcal{T}^\kappa(V) = (1-\kappa)\sum_{h=0}^{\infty}\kappa^h\mathcal{T}^{h+1}V\}$, a policy that is greedy w.r.t. a κ geometric interpolation of all multi-step greedy policies. Cf. Grill et al. [2020], relaxing the hard greedification, and adding approximations over parametric policy classes, forward search algorithms at scale, can be written as the solution to a regularized optimal control problem, by replacing the Q-values in the actor-critic regularized policy improvement objective with those resulting from lookahead search—using simulators [Silver et al., 2016a, 2017], or models [Schrittwieser et al., 2019].

In **meta-gradient based policy optimization** [Xu et al., 2018, Zahavy et al., 2020, Flennerhag et al., 2021], the policy may use a parametrized algorithm $\pi_{\theta_{t+1}} = \varphi_{\pi_{\theta_t}}(\eta_t)$, with $\varphi_{\pi_{\theta_t}}(\eta_t)$ the policy learner's algorithm, and $\eta \in \mathbb{R}^m$ meta-parameters. For computational tractability, we apply inductive biases to limit the functional class of algorithms the meta-learner searches over, e.g., to GA parameter updates $\theta_{t+1} = \theta_t + g_{\eta_t}$ The meta-parameters η can represent, e.g., inializations [Finn et al., 2017], losses [Sung et al., 2017, Wang et al., 2019, Kirsch et al., 2019, Houthooft et al., 2018, Chebotar et al., 2019, Xu et al., 2020], internal dynamics [Duan et al., 2016], exploration strategies [Gupta et al., 2018, Flennerhag et al., 2021], hyperparameters [Veeriah et al., 2019, Xu et al., 2018, Zahavy et al., 2020], and intrinsic rewards [Zheng et al., 2018]. The meta-learner's objective is to adapt the parametrized optimization algorithm based on the learner's post-update performance—unknown in RL, and replaced with a surrogate objective. Zahavy et al. [2020] uses a linear model, whereas Flennerhag et al. [2021] a quasi-Newton method [Martens, 2014] by means of a trust region with a hard cut-off after h parameter updates.

3 An operator view on optimistic policy optimization

We unify seemingly distinct *adaptive optimistic algorithms* with the following distinctive characteristics. (i) The algorithms update two functions, a policy and a Q-function. (ii) Their updates average online local solutions using *proximal (soft)* operators. This is justified, cf. Bhandari and Russo [2019], since the policy optimization is solved iteratively, by breaking it down into local subproblems, and solving those sequentially. (iii) The local solutions to these subproblems are optimistic improvements, based on *predictions* bootstrapped from the opposite sequence.

This section starts by presenting the simplest policy optimization algorithm with these properties, for finite MDPs, and a convex policy class (analyzed theoretically in Sec. 4). We then describe small changes that relax some assumptions to obtain policy gradient methods for parameterized softmax policy classes, we gradually transition to more practical and scalable policy gradient methods adaptive via meta-gradient learning.

Adaptive optimistic policy iteration (AOPI) Consider finite state and action MDPs, and a convex policy class consisting of all stochastic policies $\Pi = \{\pi \in \mathbb{R}_+^{|\mathcal{S}|\times|\mathcal{A}|} \big| \sum_{a\in\mathcal{A}} \pi(s,a) = 1, \forall s \in \mathcal{S}\}$. We express the optimistic improvement step as generating a policy with a proximal (soft) operator $\pi_{t+1} \in \mathcal{G}^\kappa_{\pi_t}(Q_t)$—a κ-interpolation ($\kappa \in (0,1]$) between the previous solution $\pi_t \in \Pi$, and best-response greedification w.r.t. a *prediction* Q_t[1], s.t. $\pi^+_{t+1} \in \mathcal{G}(Q_t) \equiv \{\pi | \mathcal{T}_\pi Q_t \geq \mathcal{T}Q_t\}$ (with possible extensions to multi-step greedification [Efroni et al., 2018a,b])

$$\pi_{t+1} = (1-\kappa)\pi_t + \kappa\pi^+_{t+1} \quad \pi^+_{t+1} = \arg\max_{\pi\in\Pi}\langle Q_t, \pi\rangle_{d_{\pi_t}} \qquad \textit{(optimistic improvement)} \tag{1}$$

Adaptation in our formalism means updating the *prediction* Q_t (and the resulting policy π_{t+1}), in hindsight, with a proximal operator $\mathcal{P}^\lambda_{\pi_{t+1}}Q_t$—a λ-interpolation ($\lambda \in (0,1]$) between the previous solution Q_t, and the most recent evaluation of the performance of the hypothesis, denoted Q^+_t, and called *"hint"*[2]

$$Q_{t+1} \in \mathcal{P}^\lambda_{\pi_{t+1}}Q_t \equiv (1-\lambda)Q_t + \lambda Q^+_{t+1} \quad \text{e.g., } Q^+_{t+1} = \mathcal{T}_{\pi_{t+1}}Q_t \qquad \textit{(hindsight adaptation)}$$

Other options for Q^+_{t+1} include ground truth from an oracle ($Q_{\pi_{t+1}}$), or approximations thereof—e.g, stochastic returns, or bootstrapped partial evaluations with $\mathcal{T}_{\pi_{t+1}}$, $\mathcal{T}^h_{\pi_{t+1}}$, or $\mathcal{T}^\lambda_{\pi_{t+1}}$. Target policies are optimal in hindsight $\pi^+_{t+2} = \arg\max_{\pi_\Pi}\langle Q_{t+1}, \pi\rangle_{d_{\pi_{t+1}}}$.

Natural actor-critics & forward search Consider a parameterized softmax policy class $\pi_\theta \in \Pi_\Theta$, with parameter vector θ, and $Q_\eta \in \mathcal{F}_\eta$, with parameter vector η. We translate the proximal update in

[1] Soft policy improvement [Kakade and Langford, 2002] uses this step with $Q_t \equiv Q_{\pi_t}$.

[2] in analogy with online convex optimization literature

Policy learner	Policy alg.	Grad learner	Grad alg.	Algorithmic instances	PS
proximal(κ)	CPI♣	best-resp($\lambda=1$)	oracle	soft PI[1]	✗
best-resp($\kappa=1$)	PI	proximal(λ)	prox DP♦	OPI[2]	✗
proximal(κ)	CPI♣	proximal(λ)	prox DP♦	AOPI	✗
proximal(κ)	MD♢	best-resp($\lambda=1$)	TD	TRPO[3], MDPO[4], MPO[5],PPO[6]	✓
best-resp($\kappa=1$)	EGD♡	proximal(λ)	AO-FTRL♥	AAPI[7], NPG[8],	✗
proximal(κ)	GD	proximal(λ)	AO-FTRL♥	STAX[9], BMG[10],	✓
proximal(κ)	OMD♠	proximal(λ)	AO-FTRL♥	MuZero[11], Muesli[12]	✓

Table 1: Adaptive and/or optimistic algorithms—characterizing (i) *optimism*—based on whether they use bootstrapped predictions in their improvement steps (indicated by the **policy & gradient algorithms** they use), and (ii) *adaptation*—whether they adapt based on their past solutions (κ, λ) or use the most recent solution as best response $(\kappa=1, \lambda=1)$. **Policy learner: ♣**—soft updates cf. Conservative Policy Iteration [Kakade and Langford, 2002], ♠ —Optimistic Mirror Ascent/Descent [Rakhlin and Sridharan, 2013b], ♢—Inexact Mirror Ascent/Descent [Beck and Teboulle, 2003], ♡ —exponentiated gradient descent. **Gradient learner: ♦**—proximal/extrapolated updates cf. [Bertsekas, 2012], **♥**—Adaptive-Optimistic Follow-the-Regularized-Leader [Mohri and Yang, 2015]. Various well-known algorithmic instantiations placed in this context: [1] Kakade and Langford [2002], [2] Bertsekas [2012], [3] Schulman et al. [2015], [4] Tomar et al. [2020], [5] Abdolmaleki et al. [2018], [6] Schulman et al. [2017], [7] Hao et al. [2021], [8] Kakade [2001], [9] Zahavy et al. [2020], [10] Flennerhag et al. [2021], [11] Schrittwieser et al. [2019], [12] Hessel et al. [2021]. Last column (**PS**) indicates whether the algorithm allows for a **p**ractical and **s**calable implementation.

Eq.1, cf. Geist et al. [2019], Bhandari and Russo [2021], to a proximal policy improvement operator, formulated as the solution to the regularized optimization problem

$$\pi_{\theta_{t+1}} = \arg\max_{\pi_\theta \in \Pi_\Theta} \langle Q_{\eta_t}, \pi \rangle_{d_{\pi_t}} - \frac{1}{\alpha} \mathrm{KL}_{[d_{\pi_t}]}(\pi_\theta, \pi_t) \quad \textit{(optimistic improvement \& projection)} \quad (2)$$

with $\alpha = \frac{1-\kappa}{\kappa}$ a (possibly time-dependent) step size, and the search, despite written over policies, can be over parameters. If $\kappa = 0$, then $\alpha \to \infty$, and we recover best-response greedification. If $\kappa = 1$ then $\alpha \to 0$ and there is no change in policies from t to $t+1$.

For adaptation, we could use a projection of the target $\tilde{Q}_t \equiv (1-\lambda)Q_{\eta_t} + \lambda Q^+_{t+1}$, with $Q^+_{t+1} \equiv \mathcal{T}_{\pi_{\theta_{t+1}}} Q_{\eta_t}$ on the representable function space $\mathcal{F}_\eta$, using the "compatible" function approximation [Kakade, 2001, Sutton et al., 1999], and recover Q-NPG, and NAC [Kakade, 2001], for $\lambda = 1$

$$Q_{\eta_{t+1}} = \arg\min_{\{Q_\eta \equiv \eta^\top \nabla_\theta \log \pi^\top_{\theta_t}\}} \mathbb{E}_{A\sim\pi_{\theta_t}, S\sim d_{\theta_t}} \left[(\tilde{Q}_t(S,A) - Q_\eta(S,A))^2 \right]$$

Solving this is impractical, since $\eta_{t+1} = \mathbf{F}^\dagger(\theta_t)\mathbb{E}_{\pi_{\theta_t}, d_{\theta_t}}[\nabla_\theta \log \pi_{\theta_t}(S,A)\tilde{Q}_t(S,A)]$, with $\mathbf{F}^\dagger(\theta_t)$ the Moore-Penrose pseudoinverse of the Fisher information matrix given by $\mathbf{F}(\theta_t) \equiv \mathbb{E}_{\pi_{\theta_t}, d_{\theta_t}}[\nabla_\theta \log \pi_{\theta_t}(A|S) \nabla_\theta \log \pi^\top_{\theta_t}(A|S)]$.

Optimistic meta-learning Meta-gradient algorithms further relax the optimistic policy improvement step to a parametric update rule $\pi_{\theta_{t+1}} \equiv \varphi_{\pi_{\theta_t}}(\eta_t)$, e.g., $\theta_{t+1} = \theta_t + g_{\eta_t}$, when limited to a functional class of parametric GA update rules $g_\eta \in \mathcal{F}_\eta$. These algorithms implement adaptation in a practical way, they project policy targets ahead of $\pi_{\theta_{t+1}}$, e.g., $\tilde{\pi}_{t+1} \propto \pi_{\theta_{t+1}} \exp(\tilde{Q}_t - Q_{\eta_t})$, with $\tilde{Q}_t \equiv (1-\lambda)Q_{\eta_t} + \lambda Q^+_{t+1}$

$$g_{\eta_{t+1}} = \arg\min_{g_\eta \in \mathcal{F}_\eta} \mathrm{KL}_{[d_{\pi_{t+1}}]}(\pi_{\theta_{t+1}}, \tilde{\pi}_{t+1}) \qquad \textit{(hindsight adaptation \& projection)}$$

Another choice (cf. Flennerhag et al. [2021]) is a parametric policy target $\tilde{\pi}_{t+1} \equiv \pi_{\tilde{\theta}_{t+h}}$, with $\tilde{\theta}_{t+1} = \theta_{t+1}$, and $\tilde{\theta}_{t+k+1} = \tilde{\theta}_{t+k} + g^+_{t+k}, \forall k \in (1..h-1)$, with g^+_{t+k} representing "hints", from experts, or bootstrapped targets. In Sec. 5, we choose an adaptive optimistic PG algorithm from this family and make certain design choices that bring it closer to practical applicability. We then analyze several properties related to optimism empirically in Sec.6.

4 Theoretical analysis

We now analyze the simplest algorithm, introduced at the beginning of Sec. 3, that is both optimistic via bootstrapped predictions, and adaptive in its use of functional averaging over past solutions. We are in the case of finite states and actions, and a convex policy class with a natural policy representation. Proofs and derivations are deferred to the Appendix.

Definition 1. (Adaptive Optimistic Policy Iteration (AOPI)) Let $\{\pi_t, Q_t\}_{t\geq 0}$, be the policy and Q-fn. sequences, $\mathcal{P}^\lambda$ a proximal operator, $\mathcal{G}(Q)$ the greedy set of Q. The Q-fn sequence adapts using "hints" $\{Q^+_t\}_{t\geq 0}$—partial policy evaluations $Q^+_{t+1} \equiv \mathcal{T}_{\pi_{t+1}} Q_t$. The two sequences evolve as

$$\pi_{t+1} \in \mathcal{G}^\kappa_{\pi_t}(Q_t) \equiv \{\pi | \pi = (1-\kappa)\pi_t + \kappa\pi^+_{t+1} \quad \pi^+_{t+1} \in \mathcal{G}(Q_t) \iff \mathcal{T}_{\pi^+_{t+1}} Q_t \geq \mathcal{T} Q_t\}$$

$$Q_{t+1} = \mathcal{P}^\lambda_{\pi_{t+1}} Q_t \equiv (1-\lambda)Q_t + \lambda Q^+_{t+1} \qquad Q^+_{t+1} \equiv \mathcal{T}_{\pi_{t+1}} Q_t$$

Observation 1. ***(Generalization of soft & optimistic PI)*** *Setting $\lambda = 1$ and using complete evaluations for the hints $Q^+_t \equiv \mathcal{T}^\infty_{\pi_t} Q_t = Q_{\pi_t}, \forall t \geq 0$, AOPI recovers soft PI ♣ [Kakade, 2001]—i.e. the policy learner uses a proximal (soft) update (κ), cf. Table 1. Using $\kappa = 1$, AOPI recovers optimistic PI ♦ [Bertsekas and Tsitsiklis, 1996]—i.e. the value learner uses a proximal (soft) update, cf. Table 1.*

The next theorem states the convergence of AOPI.

Theorem 1. ***(Convergence of AOPI)*** *Let $\{\pi_t, Q^+_t, Q_t\}_{t\geq 0}$ be the sequences generated by AOPI—Def. 1. For $|\mathcal{P}^\lambda_{\pi_1} Q_0 - Q_0| \leq \epsilon$, the sequence of Q-fns converges linearly $\lim_{t\to\infty} \|Q^* - Q_t\| = 0$, at the rate $\mathcal{O}(\xi^t)$, with $\gamma \leq \xi \equiv 1 - \lambda + \gamma\lambda$.*

Theorem 2. ***(Error propagation)*** *Consider the following relaxation of Def. 1, which uses approximations $\pi^+_{t+1} \in \mathcal{G}_{\epsilon'_{t+1}}(Q_t) \equiv \{\pi : \mathcal{T}_\pi Q_t \geq \mathcal{T} Q - \epsilon'_{t+1}\}$ and $Q^+_{t+1} \equiv \mathcal{T}_{\pi_{t+1}} Q_t + \epsilon_{t+1}$. The following error bound captures the convergence of approximate AOPI*

$$V^* - V_{\pi_t} \leq \nu^t \|Q^* - Q_0\| + \frac{\nu^t}{(1-\gamma)} \|b_0\|_{\mathbf{1}} + \sum_{k=0}^{t-1} \nu^{t-k-1} \left(\lambda \|\epsilon_k\| \mathbf{1} + \frac{(1+\lambda(1-\gamma))\kappa}{(1-\gamma)} \|\epsilon'_{k+1}\| \mathbf{1} \right)$$

for $\nu \equiv ((1-\lambda\kappa) + \kappa\lambda\gamma)$ and $b_0 = Q_0 - \mathcal{T}_{\pi_1} Q_0$.

Interpretation Online averaging with $\mathcal{P}^\lambda_\pi$ slows down the convergence through the step size λ, which averages the partial evaluations Q^+. The step sizes λ and κ also smoothen the approximation and greedification errors—ϵ and ϵ'.

Further acceleration is possible with multi-step policy improvement, cf. Efroni et al. [2018a,b], in which case $\pi^+_{t+1} \in \mathcal{G}(\mathcal{T}^\kappa Q_t) = \{\pi | \mathcal{T}^\kappa_\pi Q_t \geq (1-\kappa) \sum_{h=0}^\infty \kappa^h \mathcal{T}^{h+1} Q_t\}$. Generally this is only possible with model-based search procedures [Efroni et al., 2018a], but multi-step optimism in meta-gradient algorithms [Flennerhag et al., 2021] could be viewed as a relaxation of this procedure.

Table 1 shows how existing algorithms can be mapped into the framework by selecting an online convex optimization (OCO) [Hazan, 2017] algorithm for the policy and Q-fn learners, and maps back to the prototypical AOPI algorithm in Def. 1 (via κ and λ). This complements Table 2 in Appendix A showing convergence properties for some of these algorithms.

5 Towards a practical adaptive optimistic policy gradient algorithm

The analysis in Sec. 4 deals with the natural policy representation, and a policy class consisting of all stochastic policies. More commonly used in practice are softmax parametrizations, and parametric gradient-critics (e.g., Espeholt et al. [2018], Schulman et al. [2015, 2017], Abdolmaleki et al. [2018], Tomar et al. [2020], Zahavy et al. [2020], Flennerhag et al. [2021], Hessel et al. [2021]). We use this section to design a practical version of an adaptive optimistic PG algorithm via meta-gradient learning, starting from the generic blueprint in Alg. 1, and then empirically analyze its properties in Sec. 6.

Algorithm 1 Meta-gradient optimization for *adaptive optimistic policy gradients*

input: init params (θ_0, η_0)
input: targets $\{\tilde{\pi}_t\}_{t\geq 0}$
$S_0 \sim$ env()
for each step $t = 1, 2 \ldots$ **do**
  $A_t \sim \pi_{\theta_t}(S_t), R_t, \gamma, S_{t+1} \sim$ env(A_t)
  ———*optimistic improvement*———
  $\pi_{\theta_{t+1}} = \varphi_{\pi_{\theta_t}}(\eta_t)$ $(\theta_{t+1} = \theta_t + \beta g_{\eta_t}(S, A))$
  ———*hindsight adaptation*———
  $\eta_{t+1} = \eta_t - \zeta \nabla_{\eta_t} \mathrm{KL}(\pi_{\theta_{t+1}}(S), \tilde{\pi}_{t+1}(S))$
  if S_t is terminal: $S_t \sim$ env()
end for

Strategy We replace the standard critic with an adaptive meta-learner, which distills future targets $\tilde{\pi}_{t+1}$ into a parametric gradient-critic Q_{η_t}. We experiment with various forms of targets, and observe the effect different kinds of optimism have on the policy optimization.

Optimistic prediction Alg. 1 parametrizes the policy gradient field g_η with meta-parameter vector η. We use η to represent an adaptive *gradient-critic*—though its actual role emerges and evolves based

on its learning strategy. The adaptive optimistic policy gradient algorithm follows in expectation, the predicted gradient ascent direction $\mathbb{E}_{(S,A)\sim d_{\pi_\theta}}\left[g_\eta\right] = \mathbb{E}_{(S,A)\sim d_{\pi_\theta}}\left[\nabla_\theta \log \pi_\theta \mathrm{A}_\eta(S,A)\right]$, with advantage $\mathrm{A}_\eta(S,A) = Q_\eta(S,A) - \mathbb{E}_{\pi_\theta}\left[Q_\eta(S,A)\right]$ to update a parametric policy, using one, or more steps of GA with the parameter update rule in line 7 of Alg. 1.

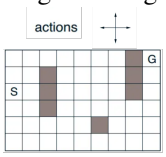
Hindsight update In contrast to standard actor-critic methods, we *meta-learn the parameter vector η—of the parametrized policy gradient—g_{η_t}, in hindsight, using one, or more steps of meta-gradient descent on the meta-objective in line 9 of Alg. 1, to move a policy's parameters—θ_t, toward post-update policies $\pi_{\theta_{t+1}}$, with higher performance $V_\rho(\pi_{\theta_{t+1}})$*. The meta-objective we use to approximate the (unknown) post-update policy performance, $V_\rho(\pi_{\theta_{t+1}})$, is a regularized linear model, cf. line 9 of Alg. 1, where the character of the trust region is induced by the choice of target policy, or gradient hint, discussed next.

Targets/hints In Sec.6, we empirically study: (i) the effect of grounded meta-optimization targets based on supervised learning—equivalent to online imitation from experts, and (ii) with self-supervised, inaccurate target predictions—obtained from a separate standard critic. Bootstrapping the meta-optimizer on itself quickly diverges without proper grounding. We leave the exploration of other ways of adding partial feedback to ground the bootstrap targets to future work.

6 Illustrative empirical analysis

In this section, we investigate the role and properties of optimism in online policy optimization, in an illustrative task. We begin with a brief study on the effect of the lookahead horizon on the optimistic step, in Sec. 6.1 to understand optimism in multi-step forward search algorithms, and distinguish between two notions of optimism. Thereafter, in Sec. 6.2, we consider the optimistic policy-gradient algorithm, trained with meta-gradient learning we introduced in Sec. 5, and investigate emerging properties for several choices of targets.

Experimental setup In both experiments, we use the discrete navigation task from [Sutton and Barto, 2018], illustrated aside (details in Appendix C).



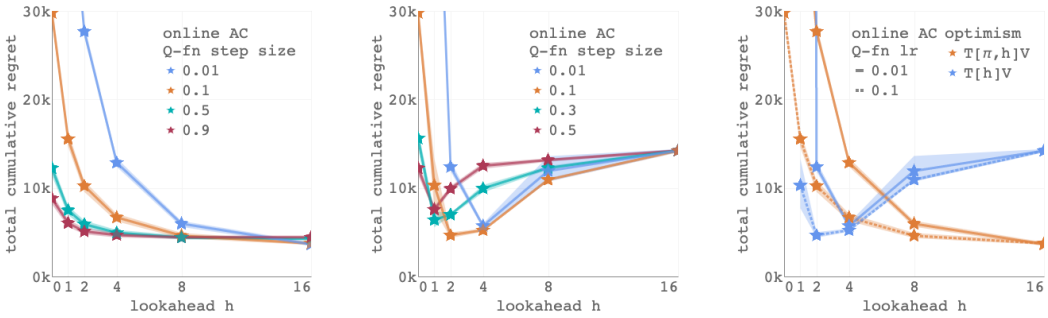
6.1 Optimism with multi-step forward search

For this illustration, we remove errors from the breadth truncation of the gradient prediction, using exact simulators, full search and no adaptation, and the only source of gradient inaccuracy is the depth truncation using a fixed limited lookahead horizon h. We use this experiment to understand the difference between: (i) optimism within the local policy evaluation problem ($\mathcal{T}^h_{\pi_t} Q_t$), and (ii) optimism within the global maximization problem ($\mathcal{T}^h Q_t$).

Algorithms We consider an online AC algorithm, with forward planning up to horizon h to compute the search Q-values $Q^h \equiv \mathcal{T}^h_{\pi_b} Q_{w_t}$, bootstrapping at the leaves on Q_{w_t}, a standard critic learned with online TD(0), and π_b, a tree-search policy. We keep a policy π_θ trained online on the objective in Eq 30, equivalent to the GA parameter update: $\theta_{t+1} = \theta_t + \beta \nabla_\theta \log \pi_{\theta_t}(A|S)\left[Q^h(S,A) - \mathbb{E}_{\pi_{t+1}}\left[Q^h(S,A)\right]\right]$ with actions sampled online from the policy π_{θ_t} at the previous iteration.

Can an online policy gradient algorithm accelerate by adapting the optimistic lookahead horizon? We use a multi-step operator in the optimistic step, which executes a value lookahead expansion up to horizon h, and tree back-up. For the tree policy π_b, we distinguish between: (i) extra policy evaluation steps with the previous policy, $Q^h \equiv \mathcal{T}^h_{\pi_t} Q_{w_t}$ (Fig 2(a)), and (ii) extra greedification steps, $Q \equiv \mathcal{T}^h Q_{w_t}$ (Fig 2(b)). The policy & Q-fn learners are trained online with $\theta_{t+1} = \theta_t + \beta g_t$, s.t. $\mathbb{E}_{d_{\pi_t},\pi_t}[g_t] \equiv \mathbb{E}_{d_{\pi_t},\pi_t}[\mathrm{A}^h(S,A)\nabla_\theta \log \pi_{\theta_t}(A|S))]$, with $\mathrm{A}^h(s,a) = Q^h(S,A) - \mathbb{E}_{\pi_t}[Q^h(S,A)]$, the advantage function using search Q-values Q^h, and critic parameters w with $w_{t+1} = w_t - \chi\delta(S,A)\nabla_w Q_{w_t}(S,A)$, and $\delta(S,A) \equiv r(S,A) + \gamma Q^h(S,A) - Q_{w_t}(S,A)$-TD error using as targets the search Q-values Q^h.

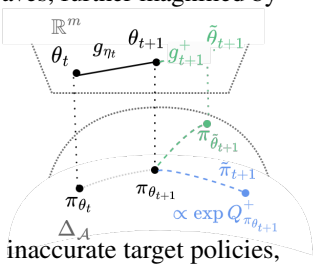
Results & observations Fig. 1(c) shows the difference between optimistic improvement—the gradient prediction has foresight of future policies on the optimization landscape, and optimistic evaluation—the gradient prediction is a refinement of the previous gradient prediction toward the optimal solution to the local PI problem. As Fig. 1(a) depicts, more lookahead steps with optimistic evaluation, can significantly improve inaccurate gradients, where accuracy is quantified by the choice of χ, the Q-fn step size for w. Thus, for $\pi_b \equiv \pi_t$, increasing $h \to \infty$, takes the optimistic step toward the exact (functional) policy gradient of the previous policy, $Q^h = \mathcal{T}^h_{\pi_b} Q_{w_t} = \mathcal{T}^h_{\pi_t} Q_{w_t} \overset{h\to\infty}{\longrightarrow} Q_{\pi_t}$. As



(a) Optimistic evaluation (b) Optimistic improvement (c) Difference between (a) and (b)

Figure 1: Optimism with extra steps of forward search with a simulator. x-axis: lookahead horizon h; y-axis: total cumulative regret $\sum_{i \leq t} V_\rho(\pi^*) - V_\rho(\pi_i)$. Lookahead targets $r + \gamma Q^h$ are used for: **(a) Optimistic evaluation** $Q^h \equiv \mathcal{T}^h_{\pi_t} Q_{w_t}$. Increasing the optimistic lookahead horizon h helps, and a horizon $h = 0$ is worst. Colored curves denote the step size χ used to learn the parameter vector w of Q_w with online TD(0). The step size controls the quality of the gradient through the accuracy of the search Q-values Q^h (more details in the main text). Shades denote confidence intervals over 10 runs. **(b) Optimistic improvement** $Q^h \equiv \mathcal{T}^h Q_{w_t}$. Intermediate values of the optimistic lookahead horizon h trade off accumulating errors for shorter horizons. **(c) Comparison between the two notions of optimism** local—evaluation within the current prediction problem, and global—improvement within the optimization problem, for two step sizes.

Fig. 1(b) shows, the optimal horizon value for optimistic improvement is another, one that trades off the computational advantage of extra depth of search, if this leads to accumulating errors, as a result of depth truncation, and bootstrapping on inaccurate values at the leaves, further magnified by greedification.



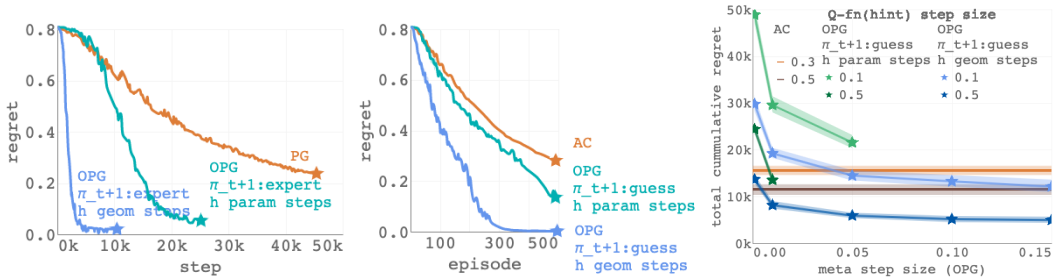
6.2 Adaptive optimistic policy gradients via meta-gradient learning

We now empirically analyze some of the properties of the practical meta-gradient based adaptive optimistic policy gradient algorithm we designed in Sec. 5.

(i) Can learned optimistic policy gradient algorithms accelerate convergence? We first remove any confounding factors arising from tracking inaccurate target policies, and resort to first-order *oracle experts* $\tilde{\pi}_{t+1}$, one update step ahead ($h = 1$) on the optimization landscape, obtained using true gradients of the post-update performance of $\pi_{\theta_{t+1}}$, $\nabla_{\pi_{\theta_{t+1}}} V_\rho(\pi_{\theta_{t+1}})$, but distinguish between two kinds of lookahead steps: (a) *parametric*, or (b) *geometric* (cf. computational graph depiction above). This difference is indicative of the farsightedness of the optimistic prediction. In particular, this distinction is based on the policy class of the target, whether it be a (a) parametric policy target $\tilde{\pi}_{t+1} \equiv \pi_{\tilde{\theta}_{t+1}}$, one step ahead, s.t. $\tilde{\theta}_{t+1} = \theta_{t+1} + \beta g^+_{t+1}$, with g^+_{t+1}, *hint*, or a (b) non-parametric policy target, $\tilde{\pi}_{t+1} \propto \pi_{\theta_{t+1}} \exp \alpha Q^+_{t+1}$, with $Q^+_{t+1} \equiv Q_{\pi_{\theta_{t+1}}}$, *hint*, for α, the step-size/temperature hyperparameter denoting the degree of greedification in the target (results are for $\alpha = 1$; $\alpha \to \infty$ means best-response greedification).

Results & observations When the meta-optimization uses an adaptive optimizer (Adam [Kingma and Ba, 2015]), Fig 2(a) answers the question in the affirmative for expert targets $\tilde{\pi}_{t+1}$ one step ahead of the learner parametric, or geometric. The large gap in performance between the two optimistic algorithms owes to the fact that target policies that are one geometric step ahead correspond to steepest directions of ascent, and consequently, may be further ahead of the policy learner in the space of parameters, leading to acceleration. Additional results illustrating sensitivity curves to hyperaparameters are added in Appendix C. When the meta-optimization uses SGD, performance of the meta-learner algorithms is slower, lagging behind the PG baseline, but the ordering over the optimistic variants is maintained (Fig 6(a) in Appendix C), which indicates that the correlation between acceleration and how far ahead the targets are on the optimization landscape is independent of the choice of meta-optimizer.

(ii) Does target accuracy impact acceleration? Next, we relax the setup from the previous experiment, and replace the oracle experts for inaccurate target predictions. Instead of the actual future gradients $\nabla_{\pi_{\theta_{t+1}}} V_\rho(\pi_{\theta_{t+1}})$, the targets now use estimates of (future) gradients of the post-update



(a) Expert targets (b) Target predictions (c) Hyperparameter sensitivity

Figure 2: Policy optimization with adaptive optimistic policy gradients (a) x-axis: number of steps, y-axis: regret $V_\rho(\pi^*) - V_\rho(\pi_t)$. Different colored curves denote: standard PG, *adaptive optimistic policy gradient learning algorithms* - with parametric target policies, and non-parametric target policies, trained with meta-gradients from expert targets. **(b)** x-axis: number of episodes, y-axis: regret. *Optimistic policy gradients* are meta-learned from target predictions. **(c)** Hyper-parameter sensitivity curves. x-axis: meta-learning rate β, y-axis: total cumulative regret $\sum_{i \leq t} V_\rho(\pi^*) - V_\rho(\pi_i)$. The plot shows *adaptive optimistic policy gradient learning algorithms* meta-learned from target predictions. Different tones depict different target accuracies, indirectly via the Q-fn step size. Straight lines show a baseline standard AC. Shades denote confidence intervals over 10 runs.

performance $\mathbb{E}_{d_{\pi_{\theta_{t+1}}}}[Q_{\pi_{\theta_{t+1}}}] \equiv \nabla_{\pi_{\theta_{t+1}}} V_\rho(\pi_{\theta_{t+1}})$. In particular, we resort to online sampling under the distribution of the optimistic policy learner, and learn $Q_{w_{t+1}} \approx Q_{\pi_{\theta_{t+1}}}$ as a standard critic, online with TD(0), s.t. $w_{t+1} = w_t - \chi[r(S,A) + \gamma Q_{w_t}(S,A) - Q_{w_t}(S,A)]\nabla_w Q_{w_t}(S,A)$, with step size χ. With respect to the policy class of the targets we experiment with the same two choices (a) *parametric*, or (b) *non-parametric* . Targets are ahead of the optimistic learner, in (a) *parameter* steps, for the former, and geometric steps for the latter.

Results & observations Even when the target predictions are inaccurate, Fig.2(b) shows that using optimistic policy gradients distilled from these target predictions can still be useful (meta-optimization uses Adam, although promising results are in Appendix C also for meta-optimization with SGD). Non-parametric targets, ahead in the optimization, show similar potential as when using oracle experts. Fig 2(c) illustrates the total cumulative regret (y-axis) stays consistent across different levels of accuracy of the policy targets (quantified via the Q-fn step sizes (χ)), indicated by different tones for each algorithm. As expected, we observe parametric targets to be less robust to step size choices, compared to non-parametric ones, analogous the distinct effect of non-covariant gradients vs natural gradients.

7 Concluding remarks

We presented a two-step—optimism & adaptation—operator-view of policy optimization, and connected seemingly distinct classes of algorithms. We investigated the theoretical properties of an adaptive optimistic PI algorithm within our formalism, and used this understanding to design a meta-gradient based adaptive optimistic PG algorithm highlighting its properties empirically.

Related work We defer an extensive discussion on related work to the appendix. The closest in spirit to this operator-view formulation is the predictor-corrector paradigm, used also by Cheng et al. [2018]. The most similar optimistic algorithm for policy optimization is AAPI [Hao et al., 2020]. Both analyze optimism from a smooth optimization perspective, whereas we focus the analysis on Bellman-operators and PI, allowing the unification. We extend the empirical analysis of Flennerhag et al. [2021], who only focused on meta-learning hyperparameters of the policy gradient. Other meta-gradient algorithms [Sung et al., 2017, Wang et al., 2019, Chebotar et al., 2019, Xu et al., 2020] take to more empirical investigations. We focused on understanding the core principles common across methods, valuable in designing new algorithms in this space, optimistic in spirit.

Future work We left many questions unanswered, about potential acceleration, and analysis beyond the convex policy class. The scope of our experiments stops before function approximation, and bootstrapping the meta-optimization on itself. Conceptually, the idea of optimizing for future performance has applicability in lifelong learning, and adaptation in non-stationary environments [Flennerhag et al., 2021, Luketina et al., 2022, Chandak et al., 2020].

Appendix

A Convergence rates for policy gradient algorithms

Alg	Opt	Π/Π_Θ	$\nabla\hat{\mathbf{V}}_\rho$	Notes	$\mathcal{O}(\mathbf{T})$	Reference
PI	✗	Π	-	convex Π	linear	Ye [2011]
API	✗	Π	$\mathrm{R}^{\lvert S\rvert\times\lvert A\rvert}$	convex Π	linear	Scherrer [2016]
SoftPI	✗	Π	-	convex Π	linear	Bhandari and Russo [2021]
GA	✗	Π_Θ	-	log barrier reg.	$\mathcal{O}(1/\sqrt{T})$	Agarwal et al. [2019]
GA	✗	Π_Θ	-	-	$\mathcal{O}(1/T)$	Mei et al. [2020b]
GA	✗	Π_Θ	-	entropy reg	linear	Mei et al. [2020b]
PGA	✗	Π_Θ	-	-	$\mathcal{O}(1/\sqrt{T})$	Bhandari and Russo [2019]
PGA	✓	Π_Θ	$\mathrm{R}^{\lvert S\rvert\times\lvert A\rvert}$	adaptive step size	$\mathcal{O}(1/\sqrt{T})$	Cheng et al. [2018]
PGA	✗	Π_Θ	$\mathrm{R}^{\lvert S\rvert\times\lvert A\rvert}$	(non/)convex Π	$\mathcal{O}(1/\sqrt{T})$	Agarwal et al. [2019]
PGA	✗	Π_Θ	$\mathrm{R}^{\lvert S\rvert\times\lvert A\rvert}$		$\mathcal{O}(1/\sqrt{T})$	Shani et al. [2019]
PGA	✗	Π_Θ	$\mathrm{R}^{\lvert S\rvert\times\lvert A\rvert}$	entropy reg.	$\mathcal{O}(1/T)$	Shani et al. [2019]
PGA	✗	Π_Θ	$\mathrm{R}^{\lvert S\rvert\times\lvert A\rvert}$	adaptive step size	linear	Khodadadian et al. [2021]
PGA	✗	Π_Θ	-	adaptive step size	linear	Bhandari and Russo [2019]
PGA	✗	Π_Θ	$\mathrm{R}^{\lvert S\rvert\times\lvert A\rvert}$	adaptive step size	linear	Xiao [2022]
PGA	✗	Π_Θ	$\mathrm{R}^{\lvert S\rvert\times\lvert A\rvert}$	entropy reg.	linear	Cen et al. [2022]
PGA	✗	Π_Θ	-	entropy reg.	linear	Bhandari and Russo [2021]
PGA	✗	Π_Θ	$\mathrm{R}^{\lvert S\rvert\times\lvert A\rvert}$	strong reg.	linear	Lan [2022]
PGA	✗	$\Pi_{\Phi^\top\Theta}$	$\Phi^\top\eta$		$\mathcal{O}(1/T)$	Agarwal et al. [2019]
PGA	✗	$\Pi_{\Phi^\top\Theta}$	$\Phi^\top\eta$	adaptive step size	$\mathcal{O}(1/T^{2/3})$	Abbasi-Yadkori et al. [2019]
PGA	✓	$\Pi_{\Phi^\top\Theta}$	$\Phi^\top\eta$	adaptive step size	$\mathcal{O}(1/T^{3/4})$	Hao et al. [2020]
PGA	✓	$\Pi_{\Phi^\top\Theta}$	$\Phi^\top\eta$	adaptive step size	$\mathcal{O}(1/\sqrt{T})$	Lazic et al. [2021]
PGA	✗	$\Pi_{\Phi^\top\Theta}$	$\Phi^\top\eta$	geom incr. step size	linear	Chen and Theja Maguluri [2022]
PGA	✗	$\Pi_{\Phi^\top\Theta}$	$\Phi^\top\eta$	geom incr. step size	linear	Alfano and Rebeschini [2023]
PGA	✗	$\Pi_{\Phi^\top\Theta}$	$\Phi^\top\eta$	geom incr. step size	linear	Yuan et al. [2023]

Table 2: Summary of previous work on rates of convergence for policy gradient algorithms (Columns) **Alg**—algorithm used, **Opt**—whether it uses optimism, $\mathbf{\Pi}/\mathbf{\Pi_\Theta}$—policy class, $\nabla\hat{\mathbf{V}}_\rho$—the space of the gradient prediction used (if not using the true gradient of the policy performance objective, $\nabla\mathbf{V}_\rho$, in which case it is marked with $-$), $\mathcal{O}(\mathbf{T})$—iteration complexity (finite sample analysis of convergence), as a function of number of iterations $\mathbf{T}$, **Notes**—other assumptions, limitations, observations. **(Algorithm)** the following abbreviations are used: PI—policy iteration, API—approximate policy iteration, SoftPI—soft policy iteration, GA—gradient ascent, PGA—projected gradient ascent (including (inexact) natural gradient ascent, (inexact) mirror ascent, (inexact) dual-averaging, (inexact) primal-dual views). **(Policy class)** the following abbreviations are used: Π—tabular with direct/natural policy parametrization, Π_Θ—tabular with softmax policy parametrization, $\Pi_{\Phi^\top\Theta}$—log-linear policy parametrization, i.e. the softmax transform is applied on a linear parametrization, $\Phi^\top\Theta$, with Φ—the feature representation, and with corresponding linear gradient approximation over the policy's features $\nabla\hat{V}_\rho = \Phi^\top\eta$, with η—parameter vector η.

B Analysis & related work

B.1 Notation

When it is clear from the context, we may ocassionally use the following simplified notation $\nabla f(x) \equiv \nabla_x f(x)$ to shorthand the gradient or the Jacobian of a function f w.r.t. its input.

Table 3: Notation

MDP	
ρ	initial state distribution
$V_\rho : \Pi \to \mathbb{R}$	policy performance
d_π	state visitation distribution under π
γ	discount factor
Indices	
t	iteration index: $t \in \{1, ..., T\}$ (also the timestep in the online setting).
T	total number of iterations.
Policies	
$\Pi, \Pi_\Theta, \Pi_{\Phi^\top \Theta}$	tabular policy class (direct/natural parametrization), parametrized policy classes: softmax, or softmax log-linear parametrization.
Φ	feature space of $\log \pi_\theta$ for the log-linear parametrization.
$\pi^* \in \Pi$	optimal policy
$\pi_t \in \Pi$	proximal policy at time t (moving average of past solutions under the κ, or α)
$\tilde{\pi}_t \in \Pi$	target (non-parametric) policy at time t
$\pi^+_{t+1} \in \Pi$	best-response policy at time t (most recent solution)
κ	step size for policy moving averages
$\theta \in \Theta$	learner (policy) parameter (π_θ)
$\tilde{\theta} \in \Theta$	target (policy) parameter ($\pi_{\tilde{\theta}}$)
Policy operators/maps.	
$\mathcal{G}(V) \equiv \{\pi \vert \mathcal{T}_\pi V \geq \mathcal{T} V\}$	greedy set of V
$\mathcal{G}^\kappa_{\pi_t} \equiv \{(1-\kappa)\pi_t + \kappa \pi^+_{t+1}\}$	line segment connecting the previous proximal policy π_t to the best-response greedification π^+_{t+1}. With $\kappa \in [0, 1]$, $\pi^+_{t+1} \in \mathcal{G}(V_t)$
$\varphi : \Pi \times \mathcal{F} \to \Pi$	generic meta-update rule, or the practical version $\varphi : \Theta \times \mathcal{F} \to \Theta$
$\langle \cdot, \cdot \rangle_{d_\pi}$	weighted dot product
$\mathrm{KL}_{[d_\pi]}(\pi, \pi')$	weighted divergence $\mathrm{KL}(\pi, \pi') = \sum_s d_\pi(s) \sum_a \pi(a\vert s) \log \frac{\pi(a\vert s)}{\pi'(a\vert s)}$
Policy gradients (Q-fns)	
$Q_{\pi_t} \in \mathbb{R}^{\vert\mathcal{S}\vert \times \vert\mathcal{A}\vert}$	unbiased Q-fn for policy π_t
$Q_t \in \mathbb{R}^{\vert\mathcal{S}\vert \times \vert\mathcal{A}\vert}$	estimated Q-fn at iteration t (proximal, moving average of past solutions under λ)
$\tilde{Q}_t \in \mathbb{R}^{\vert\mathcal{S}\vert \times \vert\mathcal{A}\vert}$	target Q-fn at iteration $t-1$ (moving average of past solutions under λ)
$Q^+_t \in \mathbb{R}^{\vert\mathcal{S}\vert \times \vert\mathcal{A}\vert}$	"hint"—best-response Q-fn at iteration $t-1$/(most recent solution)
λ	step size for Q-fn moving averages
$\eta \in \mathcal{F}$	meta-parameter, parameter of the policy gradient (g_η or Q_η)
g_{η_t}	parametric policy gradient
g_t	unbiased (ground truth, or a stochastic sample) policy gradient
Value operators/maps.	
$\mathcal{T}_\pi Q$	standard policy eval operator $\mathcal{T}_\pi Q \equiv r_\pi + \gamma P_\pi Q$
$\mathcal{T} Q$	standard optimality operator $\mathcal{T} Q \equiv \max_\pi r_\pi + \gamma P_\pi Q$
$\mathcal{P}^\lambda \mathcal{T}_\pi Q$	proximal + improvement operator $\mathcal{P}^\lambda \mathcal{T}_\pi Q \equiv (1-\lambda) Q + \lambda \mathcal{T}_\pi Q$
$\mathcal{T}^h Q, \mathcal{T}^\kappa Q$	h-step optimality operator, κ-interpolation of all multi-step optimality operators
$\mathcal{T}^h_\pi Q, \mathcal{T}^\lambda_\pi Q$	h-step eval operator, λ-interpolation of all multi-step eval operators

B.2 Optimism in policy optimization

Problem formulation The RL problem consists in finding a policy π maximizing the discounted return—the policy performance objective: $V_\rho(\pi) \equiv \mathbb{E}_{S\sim\rho}[V_\pi(S)] = (1-\gamma)\mathbb{E}_{\pi,\rho}\big[\sum_{t\geq 0}\gamma^t R_t\big]$, where $V_\pi \in \mathbb{R}^{|\mathcal{S}|}$ is the value function, and $Q_\pi \in \mathbb{R}^{|\mathcal{S}|\times|\mathcal{A}|}$ the action-value function of a policy $\pi \in \Pi = \{\pi \in \mathbb{R}_+^{|\mathcal{S}|\times|\mathcal{A}|} | \sum_{a\in\mathcal{A}}\pi(s,a) = 1, \forall s \in \mathcal{S}\}$, s.t. $Q_\pi(s,a) \equiv \mathbb{E}_\pi\left[\sum_{t=0}^\infty \gamma^t R_t | S_0 = s, A_0 = a\right]$, and $V_\pi(s) \equiv \mathbb{E}_\pi\left[Q(s,A)\right]$.

B.2.1 Policy iteration

Policy iteration The classic **policy iteration** algorithm repeats consecutive stages of (i) one-step greedy policy improvement w.r.t. a value function estimate

$$\pi_{t+1} \in \mathcal{G}(V_{\pi_t}) = \{\pi : \mathcal{T}_\pi V_{\pi_t} = \mathcal{T} V_{\pi_t}\} \iff \pi_{t+1} = \arg\max_{\pi\in\Pi}\langle \nabla V_\rho(\pi_t), \pi\rangle = \langle Q_t, \pi\rangle_{d_{\pi_t}} \tag{3}$$

with $\mathcal{G}$ the greedy set of V_{π_t}, followed by (ii) evaluation of the value function w.r.t. the greedy policy

$$V_{\pi_{t+1}} = \lim_{h\to\infty} \mathcal{T}^h_{\pi_{t+1}} V_{\pi_t} \text{ or } Q_{\pi_{t+1}} = \lim_{h\to\infty} \mathcal{T}^h_{\pi_{t+1}} Q_{\pi_t} \tag{4}$$

Approximate policy iteration Approximations of either steps lead to approximate PI (API) [Scherrer et al., 2015], in which we replace the two steps above with

$$\pi_{t+1} \in \mathcal{G}(V_{\pi_t}) = \{\pi : \mathcal{T}_\pi V_{\pi_t} \geq \mathcal{T} V_{\pi_t} - \epsilon_{t+1}\} \tag{5}$$

with ϵ_{t+1} a greedification and/or value approximation error.

Soft policy iteration Relaxing the greedification leads to **soft policy iteration**, or conservative policy iteration [Kakade and Langford, 2002], called Frank-Wolfe by Bhandari and Russo [2021]. The minimization problem decouples across states to optimize a linear objective over the probability simplex

$$\pi_{t+1} = (1-\kappa)\pi_t + \kappa\pi^+_{t+1} \text{ with } \pi^+_{t+1} = \arg\max_{\pi\in\Pi}\langle Q_{\pi_t}, \pi\rangle_{d_{\pi_t}} \tag{6}$$

for $\kappa \in [0,1]$, a (possibly time-dependent) step size, and $\langle\cdot,\cdot\rangle_d$ a state weighting that places weight $d(s)$ on any state-action pair (s,a).

Optimistic policy iteration (OPI) [Bertsekas and Tsitsiklis, 1996] relaxes the evaluation step instead to

$$Q_{t+1} = (1-\lambda)Q_t + \lambda Q^+_{t+1}, \text{ with } Q^+_{t+1} = \mathcal{T}^h_{\pi_{t+1}} Q_t, \forall h \geq 0 \tag{7}$$

with $\lambda \in [0,1]$. Other partial evaluations are possible, such as a geometric interpolation with $\tilde{\lambda}$ (possibly different than λ) of multi-step partial evaluations: $Q^+_{t+1} \equiv \mathcal{T}^{\tilde{\lambda}}_{\pi_t} Q_t = (1-\tilde{\lambda})\sum_{h=0}^\infty \tilde{\lambda}^h \mathcal{T}^{h+1}_{\pi_{t+1}}$, with $\tilde{\lambda} \in [0,1]$.

Adaptive optimistic policy iteration (AOPI) For a convex policy class consisting of all stochastic policies $\Pi = \{\pi \in \mathbb{R}_+^{|\mathcal{S}|\times|\mathcal{A}|} \big| \sum_{a\in\mathcal{A}}\pi(s,a) = 1, \forall s \in \mathcal{S}\}$, we define AOPI as the repeated application of two interleaving operators: an optimistic policy improvement with a prediction Q_t, and a hindsight adaptation on this prediction for the next Q_{t+1}. We express the optimistic improvement step as generating a policy with a proximal (soft) operator $\pi_{t+1} \in \mathcal{G}^\kappa_{\pi_t}(Q_t)$—a κ-interpolation ($\kappa \in (0,1]$) between the previous solution $\pi_t \in \Pi$, and best-response greedification w.r.t. a *prediction* Q_t, s.t. $\pi^+_{t+1} \in \mathcal{G}(Q_t) \equiv \{\pi | \mathcal{T}_\pi Q_t = \mathcal{T} Q_t\}$

$$\pi_{t+1} = (1-\kappa)\pi_t + \kappa\pi^+_{t+1} \quad \pi^+_{t+1} = \arg\max_{\pi\in\Pi}\langle Q_t, \pi\rangle_{d_{\pi_t}} \qquad \textit{(optimistic improvement)} \tag{8}$$

Adaptation in our formalism means updating this *prediction* Q_t, in hindsight, with a proximal operator $\mathcal{P}^\lambda Q^+_{t+1}$—a λ-interpolation ($\lambda \in (0,1]$) between the previous solution Q_t, and the most recent evaluation of the performance of the hypothesis, denoted Q^+_{t+1}, and called *"hint"*[3]—options include ground truth from an oracle ($Q_{\pi_{t+1}}$), or approximations thereof—e.g, stochastic returns, or bootstrapped partial evaluations with $\mathcal{T}^h_{\pi_{t+1}} Q_t$. For simplicity, consider $h = 1$

$$Q_{t+1} \in \mathcal{P}^\lambda Q^+_{t+1} \equiv (1-\lambda)Q_t + \lambda Q^+_{t+1} \quad Q^+_{t+1} = \mathcal{T}_{\pi_{t+1}} Q_t \qquad \textit{(hindsight adaptation)} \tag{9}$$

[3] in analogy with online convex optimization literature

B.2.2 Policy gradients

Projected Gradient Descent Starting with some policy $\pi \in \Pi$, an iteration of projected gradient ascent with step size α updates to the solution of the regularized problem

$$\pi_{t+1} = \arg\max_{\pi} \langle \nabla V_\rho(\pi_t), \pi \rangle + \frac{1}{\alpha} \sum_{s \in \mathcal{S}} d_{\pi_t}(s) \sum_{a \in \mathcal{A}} (\pi(a|s) - \pi_t(a|s))^2 \tag{10}$$

$$= \arg\max_{\pi} \langle Q_{\pi_t}, \pi \rangle_{d_\pi} + \frac{1}{\alpha} \|\pi - \pi_t\|^2_{2,d_{\pi_t}} \tag{11}$$

which is a first-order Taylor expansion of V_ρ w.r.t. the policy's functional representation π (see Bhandari and Russo [2021, 2019])

$$V_\rho(\pi') = V_\rho(\pi) + \langle \nabla V_\rho(\pi), \pi' - \pi \rangle + \mathcal{O}(\|\pi' - \pi\|^2) \tag{12}$$

$$= V_\rho(\pi) + \langle Q_\pi, \pi' - \pi \rangle_{d_\pi} + \mathcal{O}(\|\pi' - \pi\|^2) \tag{13}$$

With per state decoupling, for specific values of α this yields a per state projection on the decoupled probability simplex

$$\pi_{t+1} = \mathrm{Proj}^{\Pi}_{[d_{\pi_t}]} \tilde{\pi}_{t+1} = \arg\max_{\pi \in \Pi} \|\pi - \tilde{\pi}_{t+1}\|^2_{2,d_{\pi_t}} \text{ with } \tilde{\pi}_{t+1} = \pi_t + \alpha Q_{\pi_t} \tag{14}$$

with $\|\cdot\|^2_{2,d_{\pi_t}}$ the weighted L_2-norm.

Mirror descent (MD) Mirror descent adapts to the geometry of the probability simplex by using a non-Euclidean regularizer. The specific regularizer used in RL is the entropy function $H(\pi) \equiv \pi \log \pi$, such that the resulting mirror map is the $\log$ function. The regularizer decouples across the state space and captures the curvature induced by the constraint of policies lying on the policy simplex via the softmax policy transform.

Starting with some policy $\pi_t \in \Pi_\Theta$, an iteration of mirror descent with step size α updates to the solution of a regularized problem

$$\pi_{t+1} = \arg\max_{\pi \in \Pi} \langle \nabla V_\rho(\pi_t), \pi \rangle + \frac{1}{\alpha} \sum_{s \in \mathcal{S}} d_{\pi_t}(s) \, \mathrm{KL}(\pi(s), \pi_t(s)) \tag{15}$$

$$= \arg\max_{\pi \in \Pi} \langle Q_{\pi_t}, \pi \rangle_{d_{\pi_t}} + \frac{1}{\alpha} \mathrm{KL}_{[d_{\pi_t}]}(\pi, \pi_t) \tag{16}$$

which is known to be the exponentiated gradient ascent update $\pi_{t+1} = \frac{\pi_t \exp \alpha Q_{\pi_t}}{\sum_a \pi(a|\cdot) \exp \alpha Q_{\pi_t}(\cdot, a)}$ (obtained using the Lagrange approach, see Bubeck [2015]).

Using state decoupling, for specific values of α we may also write MD as a projection using the corresponding Bregman divergence for the mirror map $\nabla_\pi H(\pi)$ (cf. Bubeck [2015])

$$\pi_{t+1} = \mathrm{Proj}^{\Pi,H}_{[d_{\pi_t}]} \tilde{\pi}_{t+1} = \arg\max_{\pi \in \Pi} \mathrm{KL}_{[d_\pi]}(\pi, \tilde{\pi}_{t+1}) \text{ with} \tag{17}$$

$$\log \tilde{\pi}_{t+1} = \log \pi_t + \alpha Q_{\pi_t} - \log \sum_a \pi(a|\cdot) \exp \alpha Q_{\pi_t}(\cdot, a) \tag{18}$$

Policy parametrization For parametric policy classes the search written over policies, translates into similar versions of the linear objective, except over policy parameters. Since the class of softmax policies can approximate stochastic policies to arbitrary precision, this is nearly (we can only come infinitesimally close to an optimal policy) the same as optimizing over the class Π.

Natural policy gradients (NPG) The natural policy gradient (NPG) of Kakade [2001] applied to the softmax parameterization is actually an instance of mirror descent for the entropy-based regularizer H.

Natural policy gradient is usually described as steepest descent in a variable metric defined by the Fisher information matrix induced by the current policy [Kakade, 2001, Agarwal et al., 2019]

$$\theta_{t+1} = \theta_t + \alpha \mathbf{F}_\rho(\theta_t)^\dagger \nabla_{\theta_t} V_\rho(\pi_{\theta_t}) \tag{19}$$

$$\mathbf{F}_\rho(\theta_t) = \mathbb{E}_{S \sim d_{\pi_{\theta_t}}, A \sim \pi_{\theta_t}} \left[\nabla_{\theta_t} \log \pi_{\theta_t} \nabla_{\theta_t} \log \pi_{\theta_t}^\top \right] \tag{20}$$

and is equivalent to mirror descent under some conditions [Raskutti and Mukherjee, 2014]. Cf. Bhandari and Russo [2021], Li et al. [2021], the aforementioned base MD and NPG updates are closely related to the practical instantiations in TRPO [Schulman et al., 2015], PPO [Schulman et al., 2017], MPO [Abdolmaleki et al., 2018], MDPO [Tomar et al., 2020]. All these algorithic instantiations use approximations for the gradient direction.

B.2.3 Actor-critic methods

Generally, in RL, an agent only has access to partial evaluations of the gradient $\nabla_\pi V_\rho(\pi)$, and commonly these involve some sort of internal representation of the action-value function $Q_t \approx Q_{\pi_t}$.

Natural actor-critic. MD with an estimated critic. Consider a parameterized softmax policy class $\pi_\theta \in \Pi_\Theta$, with parameter vector θ, and $Q_\eta \in \mathcal{F}_\eta$, with parameter vector η, s.t. For the softmax policy class, this will be $\log \pi_\theta$, for $\Pi_\Theta = \left\{\pi_\theta \middle| \pi_\theta(s,a) = \frac{\exp f_\theta(s,a)}{\sum_{a' \in \mathcal{A}} \exp f_\theta(s,a')} \forall s \in \mathcal{S}, a \in \mathcal{A}, \theta \in \mathbb{R}^m\right\}$, with f_θ a differentiable function, either tabular $f_\theta(s,a) = \theta_{s,a}$, log-linear $f_\theta(s,a) = \phi(s,a)^\top \theta$, with ϕ a feature representation, or neural (f_θ-a neural network) parametrizations [Agarwal et al., 2019].

Written as a proximal policy improvement operator, at iteration t, starting with some policy $\pi_t \equiv \pi_{\theta_t}$. the next policy is the solution to the regularized optimization problem

$$\pi_{\theta_{t+1}} = \arg\max_{\pi_\theta \in \Pi_\Theta} \langle Q_{\eta_t}, \pi_\theta \rangle_{d_{\pi_t}} - \frac{1}{\alpha} \mathrm{KL}_{[d_{\pi_t}]}(\pi_\theta, \pi_t) \quad \textit{(optimistic improvement \& projection)} \tag{21}$$

with $\alpha = \frac{1-\kappa}{\kappa}$ a (possibly time-dependent) step size. If $\kappa = 0$, then $\alpha \to \infty$, and we recover best-response greedification. If $\kappa = 1$ then $\alpha \to 0$ and there is no change in policies from t to $t+1$.

Using the connection between the NPG update rule with the notion of compatible function approximation [Sutton et al., 1999], as formalized in [Kakade, 2001], we may try to approximate the functional gradient using η

$$\mathbf{F}_\rho(\theta)^\dagger \nabla_\theta V_\rho(\pi_\theta) = \frac{\eta}{1-\gamma} \tag{22}$$

where η are parameters of an advantage function A_η—which is the solution to the projection of A_{π_θ} on the dual gradient space of π, the space spanned by the particular feature representation that uses $\phi_t \equiv \nabla_\theta \log \pi_{\theta_t}$ as (centered) features

$$\eta_t = \arg\min_\eta \mathbb{E}_{S \sim d_{\pi_{\theta_t}}, A \sim \pi_{\theta_t}} [(\eta^\top \phi_t(S,A) - \mathrm{A}_{\pi_{\theta_t}}(S,A))^2] \tag{23}$$

Similarly there is an equivalent version for Q-NPG considering possibly (un-centered) features ($\phi_{s,a}$, for $f_\theta(s,a) = \phi_{s,a}^\top \theta$) and projecting

$$\eta_t = \arg\min_\eta \mathbb{E}_{S \sim d_{\pi_{\theta_t}}, A \sim \pi_{\theta_t}} [(\eta^\top \phi_t(S,A) - Q_{\pi_{\theta_t}}(S,A))^2] \tag{24}$$

For both of them we can now replace the NPG parameter update with

$$\theta_{t+1} = \theta_t + \alpha \eta_t \tag{25}$$

Adaptive optimistic policy gradients We may write a generalization of the NAC, which keeps the same policy improvement with parameters η, or the MD policy improvement with an estimated critic Q_η, except we change the critic's target to a proximal target $\tilde{Q}_t \equiv (1-\lambda) Q_{\eta_t} + \lambda Q^+_{t+1}$, with Q^+_{t+1}—hints, e.g., $Q^+_{t+1} = (\mathcal{T}_{\pi_{t+1}} Q_{\eta_t})^h$. We recover Q-NPG, and NAC [Kakade, 2001], for $\lambda = 1$.

For intermediate values of λ, we may write the update as a proximal update, for $Q_{\eta_t} = \eta^\top \phi(S,A)$

$$Q_{\eta_{t+1}} = \arg\min_\eta \mathbb{E}_{A \sim \pi_{\theta_t}, S \sim d_{\theta_t}} \left[(\tilde{Q}_t(S,A) - Q_\eta(S,A))^2 \right] \tag{26}$$

Solving this is impractical, since $\eta_{t+1} = \mathbf{F}^\dagger(\theta_t) \mathbb{E}_{\pi_{\theta_t}, d_{\theta_t}} [\nabla_\theta \log \pi_{\theta_t}(S,A) \tilde{Q}_t(S,A)]$, with $\mathbf{F}^\dagger(\theta_t)$ the Moore-Penrose pseudoinverse of the Fisher information matrix given by $\mathbf{F}(\theta_t) \equiv \mathbb{E}_{\pi_{\theta_t}, d_{\theta_t}} [\nabla_\theta \log \pi_{\theta_t}(A|S) \nabla_\theta \log \pi_{\theta_t}^\top(A|S)]$.

B.2.4 Forward search

Multi-step policy iteration The single-step based policy improvement used in the aforementioned algorithms, e.g., policy iteration, approximate PI, actor-critic methods, and its practical algorithmic implementations, is not necessarily the optimal choice. It has been empirically demonstrated in RL algorithms based on Monte-Carlo Tree Search (MCTS)[Browne et al., 2012] (e.g., Schrittwieser et al. [2019], Schmidhuber [1987]) or Model Predictive Control (MPC), that multiple-step greedy policies can perform conspicuously better. Generalizations of the single-step greedy policy improvement

include (i) h-step greedy policies, and (ii) κ–greedy policies. The former output the first optimal action out of a sequence of actions, solving a non-stationary h-horizon control problem:

$$\pi(s) \in \arg\max_{\pi_0} \max_{\pi_1,\dots\pi_{h-1}} \mathbb{E}^{\pi_0,\dots\pi_{h-1}} \left[\sum_{t=0}^{h-1} \gamma^t r(S_t, \pi_t(S_t)) + \gamma^h V(S_h) | S_0 = s \right] \tag{27}$$

equivalently described in operator notation as $\pi \in \mathcal{G}(\mathcal{T}^{h-1}V) \equiv \{\pi | \mathcal{T}_\pi T^{h-1} V \geq \mathcal{T}^h V\}$. A κ-greedy policy interpolates over all geometrically κ-weighted h-greedy policies $\pi \in \mathcal{G}(\mathcal{T}^\kappa V) \equiv \{\pi | \mathcal{T}_\pi^\kappa V \geq \mathcal{T}^\kappa V, \mathcal{T}_\pi^\kappa \equiv (1-\kappa) \sum_{h=0}^\infty \kappa^h \mathcal{T}_\pi^{h+1}\}$.

Multi-step soft policy iteration Efroni et al. [2018b] shows that when using soft updates with $h > 1$

$$\pi_{t+1} = (1-\alpha)\pi_t + \alpha \pi_{t+1}^+, \pi_{t+1}^+ \in \mathcal{G}(\mathcal{T}^{h-1}V) \equiv \{\pi | \mathcal{T}_\pi T^{h-1} V \geq \mathcal{T}^h V\} \tag{28}$$

policy improvement is guaranteed only for $\alpha = 1$, and when using

$$\pi_{t+1} = (1-\alpha)\pi_t + \alpha \pi_{t+1}^+, \pi_{t+1}^+ \in \mathcal{G}(\mathcal{T}^\kappa V) \equiv \{\pi | \mathcal{T}_\pi^\kappa V \geq \mathcal{T}^\kappa V, \mathcal{T}_\pi^\kappa \equiv (1-\kappa) \sum_{h=0}^\infty \kappa^h \mathcal{T}_\pi^{h+1}\} \tag{29}$$

policy improvement is guaranteed only for $\alpha \in [\kappa, 1]$. This result appears in Efroni et al. [2018b], and a more general version in Konda and Borkar [1999].

Tree search Notable examples of practical algorithms with empirical success that perform multi-step greedy policy improvement are AlphaGo and Alpha-Go-Zero [Silver et al., 2016a, 2017, 2016b], MuZero [Schrittwieser et al., 2019]. There, an approximate online version of multiple-step greedy improvement is implemented via Monte Carlo Tree Search (MCTS) [Browne et al., 2012]. In particular, Grill et al. [2020] shows that the tree search procedure implemented by AlphaZero is an approximation of the regularized optimization problem

$$\pi_{\theta_{t+1}} = \arg\max_{\pi_\theta \in \Pi_\Theta} \langle Q_t^h, \pi \rangle_{d_{\pi_t}} - \frac{1}{\alpha_t} \mathrm{KL}_{[d_{\pi_t}]}(\pi_\theta, \pi_t) \quad \textit{(optimistic improvement \& projection)} \tag{30}$$

with Q_t^h—the search Q-values, i.e., those estimated by the search algorithm that approximates $\mathcal{T}^h Q_{w_t}$ with stochastic sampling of trajectories in a tree up to a horizon h, and bootstrapping on a Q-fn estimator at the leaves. For a full description of the algorithm, refer to Silver et al. [2017]. The step size α_t captures the exploration strategy, and decreases the regularization based on the number of simulations.

B.2.5 Meta-learning

Optimistic meta-gradients Meta-gradient algorithms further relax the optimistic policy improvement step to a parametric update rule $\pi_{\theta_{t+1}} \equiv \varphi_{\pi_{\theta_t}}(\eta_t)$, e.g., $\theta_{t+1} = \theta_t + g_{\eta_t}$, when limited to a functional class of parametric GA update rules $g_\eta \in \mathcal{F}_\eta$. These algorithms implement adaptation in a practical way, they project policy targets $\tilde{\pi}_{t+1}$ ahead of $\pi_{\theta_{t+1}}$

$$g_{\eta_{t+1}} = \arg\min_{g_\eta \in \mathcal{F}_\eta} \mathrm{KL}_{[d_{\pi_{t+1}}]}(\pi_{\theta_{t+1}}, \tilde{\pi}_{t+1}) \qquad \textit{(hindsight adaptation \& projection)} \tag{31}$$

The targets can be parametric $\tilde{\pi}_{t+1} \equiv \pi_{\tilde{\theta}_{t+h}}$, initialized from $\tilde{\theta}_{t+1} = \theta_{t+1}$, and evolving for h step further ahead of θ_{t+1}, s.t. $\tilde{\theta}_{t+k} = \tilde{\theta}_{t+k-1} + g_{t+k-1}^+, \forall k \leq h$, with g_{t+k}^+ representing "hints", from oracle experts, or bootstrapped targets—e.g., Flennerhag et al. [2021] uses the part of the hints of g_{t+k}^+ to be predictions of the meta-learner, and part of them supervised targets. Alternatively, targets may be non-parameteric, e.g., $\tilde{\pi}_{t+1} \propto \pi_{\theta_{t+1}} \exp(Q_t^+ - Q_{\eta_t})$, e.g., if $Q_t^+ = \mathcal{T}_{\pi_{t+1}} Q_{\eta_t}$ then $\tilde{\pi}_{t+1} \propto \pi_{\theta_{t+1}} \exp(\mathcal{T}_{\pi_{t+1}} Q_{\eta_t} - Q_{\eta_t}) = \pi_{\theta_{t+1}}$—capturing the advantage of using the hypothesis $\pi_{\theta_{t+1}}$.

B.2.6 Optimism in online convex optimization

One way to design and analyze iterative optimization methods is through online linear optimization (OLO) algorithms.

Online learning Policy optimization through the lens of online learning [Hazan, 2017] means treating the policy optimization algorithm as the learner in online learning and each intermediate policy that it produces as an online decision. The following steps recast the iterative process of policy optimization into a standard online learning setup: (i) at iteration t the learner plays a decision $\pi_t \in \Pi$, (ii) the environment responds with feedback on the decision π_t, and the process repeats. The iteration t might be different than the timestep of the environment. Generally, it is assumed that the learner receives an unbiased stochastic approximation as a response, whereas that is not always the case for RL agents, using bootstrapping in their policy gradient estimation with a learned value function.

For an agent it is important to minimize the **regret** after T iterations

$$\mathrm{Reg}_T \equiv \sum_{t=0}^{T-1} \left(V_\rho(\pi^*) - V_\rho(\pi_t)\right) \tag{32}$$

The goal of **optimistic online learning** algorithms [Rakhlin and Sridharan, 2013a,b, 2014] is obtain better performance, and thus guaranteed lower regret, when playing against "easy" (i.e., predictable) sequences of online learning problems, where past information can be leveraged to improve on the decision at each iteration.

Predictability An important property of the above online learning problems is that they are not completely adversarial. In RL, the policy's true performance objective cannot be truly adversarial, as the same dynamics and cost functions are used across different iterations. In an idealized case where the true dynamics and cost functions are exactly known, using the policy returned from a model-based RL algorithm would incur zero regret, since only the interactions with the real MDP environment, not the model, are considered in the regret minimization problem formulation. The main idea is to use (imperfect) predictive models, such as off-policy gradients and simulated gradients, to improve policy learning.

B.2.7 Online learning algorithms

We now summarize two generalizations of the well-known core algorithms of online optimization for predictable sequences, cf. Joulani et al. [2020]: (i) a couple variants of optimistic mirror descent [Chiang et al., 2012, Rakhlin and Sridharan, 2013a,b, Chiang et al., 2012], including extragradient descent (Korpelevich [1976], and mirror-prox [Nemirovski, 2004, Juditsky et al., 2011], and (ii) adaptive optimistic follow-the-regularized-leader (AO-FTRL) [Rakhlin and Sridharan, 2013a, 2014, Mohri and Yang, 2016].

Optimistic mirror descent (OMD). Extragradient methods Starting with some previous iterate $x_t \in \mathcal{X}$, an OMD [Joulani et al., 2020] learner x uses a prediction $\tilde{g}_{t+1} \in \mathcal{X}^*$ ($\mathcal{X}^*$—dual space of $\mathcal{X}$) to minimize the regret on its convex loss function $f : \mathcal{X} \to \mathbb{R}$ against an optimal comparator $x^* \in \mathcal{X}$ with

$$x_{t+1} = \arg\min_{x \in \mathcal{X}} \langle g_t + \tilde{g}_{t+1} - \tilde{g}_t, x\rangle + \mathcal{B}_\Omega(x, x_t) \tag{33}$$

with $\tilde{g}_{t+1} \approx \nabla f(x_{t+1})$ optimistic gradient prediction, and $g_t \equiv \nabla f(x_t)$ true gradient feedback, $\mathcal{B}_\Omega$ a Bregman divergence with mirror map Ω.

Extragradient methods consider two-step update rules for the same objective using an intermediary sequence $\tilde{x}$

$$\tilde{x}_{t+1} = \arg\min_{x \in \mathcal{X}} \langle \tilde{g}_{t+1}, x\rangle + \mathcal{B}_\Omega(x, x_t) \tag{34}$$

$$x_{t+1} = \arg\min_{\pi \in \Pi} \langle g^+_{t+1}, x\rangle + \mathcal{B}_\Omega(x, x_t) \tag{35}$$

with $\tilde{g}_{t+1} \approx \nabla f(x_{t+1})$ a gradient prediction, and $g^+_{t+1} \equiv \nabla f(\tilde{x}_{t+1})$ the true gradient direction, but for the intermediary optimistic iterate $\tilde{x}_{t+1}$.

Adaptive optimistic follow-the-regularized-leader (AO-FTRL) A learner using AO-FTRL updates x using

$$x_{t+1} = \arg\min_{x \in \mathcal{X}} \langle g_{0:t} + \tilde{g}_{t+1}, x\rangle + \omega_{1:t-1}(x) \tag{36}$$

where $g_{0:t} = \sum_{j=0}^{t} g_j$ are true gradients, $\tilde{g}_{t+1}$ is the optimistic part of the update, a prediction of the gradient before it is received, and $\omega_{0:t}(x) = \sum_{j=0}^{t} \omega_j(x)$ represent the "proximal" part of this adaptive regularization (cf. Joulani et al. [2020]), counterparts of the Bregman divergence we have for MD updates that regularizes iterates to maintain proximity.

B.2.8 Policy optimization with online learning algorithms

Cheng et al. [2018] follows the extragradient approach for policy optimization

$$\tilde{\pi}_{t+1} = \arg\max_{\pi \in \Pi} \langle Q_t, \pi \rangle + \mathrm{KL}(\pi, \pi_t) \tag{37}$$

$$\pi_{t+1} = \arg\max_{\pi \in \Pi} \langle Q_{\pi_t}, \pi \rangle - \mathrm{KL}(\pi, \pi_t) \tag{38}$$

but changes the second sequence to start from the intermediary sequence and add just a correction

$$\tilde{\pi}_{t+1} = \arg\max_{\pi \in \Pi} \langle Q_t, \pi \rangle - \mathrm{KL}(\pi, \pi_t) \tag{39}$$

$$\pi_{t+1} = \arg\max_{\pi \in \Pi} \langle Q_{\pi_t} - Q_t, \pi \rangle - \mathrm{KL}(\pi, \tilde{\pi}_{t+1}) \tag{40}$$

This approach uses $\tilde{\pi}_{t+1}$ as the optimistic prediction, and π_{t+1} as the hindsight corrected prediction—a policy optimal in hindsight w.r.t. the average of all previous Q-functions rather than just the most recent one. But it needs an additional model for the value functions Q_t, and another learning algorithm to adapt Q_t to the Q_{t+1}. Additionally, an agent does not generally have access to Q_{π_t}, but only partial evaluations.

Hao et al. [2020] also designs an adaptive optimistic algorithm based on AO-FTRL, which updates

$$\pi_{t+1} = \arg\max_{\pi \in \Pi} \left\langle \left(\sum_{j=0}^{t} Q_j + \hat{Q}_{t+1} \right), \pi \right\rangle - \alpha_t \omega(\pi) \tag{41}$$

with ω—a regularizer, and with $Q_j \approx Q_{\pi_j}, \forall j \leq t$ predictions for the true gradients, and $\hat{Q}_{t+1} \approx Q_{\pi_{t+1}}$ is also a prediction for the gradient of the next policy, which uses the previous predictions $Q_{\pi_j}, \forall j \leq t$ to compute it. The authors also propose an adaptive method for learning α_t that uses gradient errors of $Q_j, \forall j \leq t$. Averaging value functions has also been explored by Vieillard et al. [2020b] and Vieillard et al. [2020a].

Our approach is to make the prediction Q_t adaptive, based on partial evaluations of the policy π_{t+1}, $Q^+_{t+1} \equiv \mathcal{T}_{\pi_{t+1}} Q_t$, rather than complete evaluations $Q^+_{t+1} = Q_{\pi_{t+1}}$, and to (partially) correct this prediction Q_t to Q_{t+1} in hindsight via the intermediary target $\tilde{\pi}_{t+1}$, which is ahead of π_{t+1} on the optimization landscape

$$\pi_{t+1} = \arg\min_{\pi \in \Pi} \langle Q_t, \pi \rangle - \mathrm{KL}(\pi, \pi_t) \tag{42}$$

$$\tilde{\pi}_{t+1} = \arg\max_{\pi \in \Pi} \langle Q_t + \lambda \left(Q^+_{t+1} - Q_t \right), \pi \rangle - \mathrm{KL}(\pi, \pi_t) \tag{43}$$

$$Q_{t+1} = \arg\min_{Q} \mathrm{KL}(\pi_{t+1}, \tilde{\pi}_{t+1}) \iff Q_{t+1} = Q_t + \lambda \left(Q^+_{t+1} - Q_t \right) \tag{44}$$

If $\lambda = 1$ and $Q^+_{t+1} = Q_{\pi_{t+1}}$ then $Q_{t+1} = Q_{\pi_{t+1}}$. The interpretation is that Q_t is now the optimistic prediction, and Q_{t+1} is the hindsight corrected prediction, with which the next policy is computed.

B.3 Analysis

We refer to the following definition of AOPI from the main text restated below.

Definition 1. (Adaptive Optimistic Policy Iteration (AOPI)) Let $\{\pi_t, Q_t\}_{t\geq 0}$, be the policy and Q-fn. sequences, $\mathcal{P}^\lambda$ a proximal operator, $\mathcal{G}(Q)$ the greedy set of Q. The Q-fn sequence adapts using "hints" $\{Q_t^+\}_{t\geq 0}$—partial policy evaluations $Q_{t+1}^+ \equiv \mathcal{T}_{\pi_{t+1}} Q_t$. The two sequences evolve as

$$\pi_{t+1} \in \mathcal{G}^\kappa_{\pi_t}(Q_t) \equiv \{\pi | \pi = (1-\kappa)\pi_t + \kappa\pi^+_{t+1} \quad \pi^+_{t+1} \in \mathcal{G}(Q_t) \iff \mathcal{T}_{\pi^+_{t+1}} Q_t \geq \mathcal{T} Q_t\} \tag{45}$$

$$Q_{t+1} = \mathcal{P}^\lambda_{\pi_{t+1}} Q_t \equiv (1-\lambda)Q_t + \lambda Q^+_{t+1} \qquad Q^+_{t+1} \equiv \mathcal{T}^h_{\pi_{t+1}} Q_t \tag{46}$$

Observation 1. ***(Generalization of soft & optimistic PI)*** *Setting $\lambda = 1$ and using complete evaluations for the hints $Q_t^+ \equiv \mathcal{T}^\infty_{\pi_t} Q_t = Q_{\pi_t}, \forall t \geq 0$, AOPI recovers soft PI ♣ [Kakade, 2001]—i.e. the policy learner uses a proximal (soft) update (a κ-interpolation between the past solution and the most recent best-response), cf. Table 1. Using $\kappa = 1$ and $h = 1$, AOPI recovers optimistic PI ♦ [Bertsekas and Tsitsiklis, 1996]—i.e. the value learner uses a proximal (soft) update(a λ interpolation between the past solution and the most recent best-response), cf. Table 1.*

The analysis uses the following generic assumptions about the operators $\mathcal{T}_\pi$, $\mathcal{T}$

Assumption 1. (Contraction) For all Q, and π, for the functions $\mathcal{T}_\pi$, $\mathcal{T}$, and $\gamma \in (0,1)$, we have

$$\|\mathcal{T}_\pi Q - \mathcal{T}_\pi Q'\| \leq \gamma \|Q - Q'\| \tag{47}$$

Assumption 2. (Monotonicity) For all Q, and π, for the functions $\mathcal{T}_\pi$, $\mathcal{T}$, we have

$$Q \leq Q' \implies \mathcal{T}_\pi Q \leq \mathcal{T}_\pi Q' \tag{48}$$

$$\implies Q \leq Q' \implies \mathcal{T} Q \leq \mathcal{T} Q' \tag{49}$$

Consider a behaviour policy π, a target policy, $\tilde{\pi}$ or the greedy policy w.r.t. Q, namely $\pi^+ = \arg\max_{\pi\in\Pi} \langle \pi, Q\rangle_d$. The two proximal operators, for evaluation ($\mathcal{T}_\pi$) and optimality ($\mathcal{T}$), are

$$\begin{aligned}
\mathcal{P}^\lambda_{\pi,\tilde{\pi}} Q &= (1-\lambda)Q + \lambda \mathcal{T}_{(1-\kappa)\pi + \kappa\tilde{\pi}} Q \qquad (evaluation) && (50)\\
&= (1-\lambda)Q + \lambda(1-\kappa)\mathcal{T}_\pi Q + \lambda\kappa \mathcal{T}_{\tilde{\pi}} Q && (51)\\
&= Q + \lambda\left((1-\kappa)\mathcal{T}_\pi Q + \kappa \mathcal{T}_{\tilde{\pi}} Q - Q\right) && (52)
\end{aligned}$$

and

$$\begin{aligned}
\mathcal{P}^\lambda_{\pi,\pi^+} Q &= (1-\lambda)Q + \lambda \mathcal{T}_{(1-\kappa)\pi + \kappa\pi^+} Q \qquad (control) && (53)\\
&= (1-\lambda)Q + \lambda(1-\kappa)\mathcal{T}_\pi Q + \lambda\kappa \mathcal{T} Q && (54)\\
&= Q + \lambda\left((1-\kappa)\mathcal{T}_\pi Q + \kappa \mathcal{T} Q - Q\right) && (55)
\end{aligned}$$

B.3.1 Properties of $\mathcal{P}^\lambda$

Definition 2. ("On-policy" proximal operator $\mathcal{P}^\lambda_\pi Q = (1-\lambda)Q + \lambda\mathcal{T}_\pi Q$) Let $\mathcal{P}^\lambda_\pi Q$ be the interpolation between identity and the operator $\mathcal{T}_\pi$, s.t.

$$\mathcal{P}^\lambda_\pi Q = (1-\lambda)Q + \lambda\mathcal{T}_\pi Q \quad \text{(interpolation between identity \& bootstrapped improvement)} \tag{56}$$

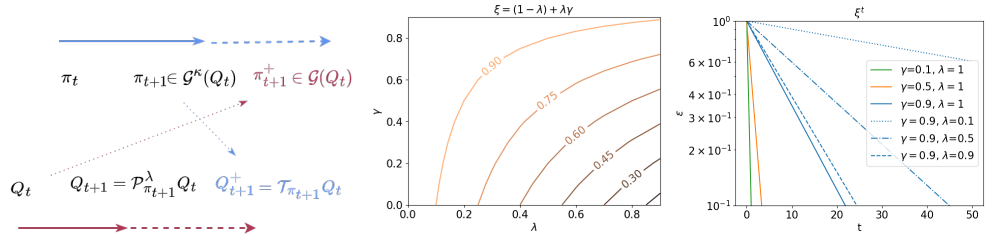
Lemma 1. ***(Contraction $\mathcal{P}^\lambda_\pi$)*** *The mapping $\mathcal{P}^\lambda_\pi$ in Def.2 is a contraction.*

Proof. For any Q, Q', and $\pi' = (1-\kappa)\pi + \kappa\pi^+$ we have the property

$$\begin{aligned}
\|\mathcal{P}^\lambda_{\pi'} Q - \mathcal{P}^\lambda_{\pi'} Q'\| &= \|((1-\lambda)(Q-Q') + \lambda\gamma P_{\pi'}(Q-Q')\| && (57)\\
&= \|((1-\lambda)(Q-Q') + \lambda\gamma((1-\kappa)P_\pi + \kappa P_{\pi^+})(Q-Q')\| && (58)\\
&\leq ((1-\lambda) + \lambda\gamma)\|Q-Q'\| && (59)
\end{aligned}$$

since for any $\kappa \in (0,1]$, the mixture $\pi' = (1-\kappa)\pi + \kappa\pi^+$ of any policy π and the one-step greedy policy π^+, is monotonically better then π [Efroni et al., 2018b].

Let $\xi \equiv (1-\lambda) + \lambda\gamma$ be the contraction coefficient. Since $\xi \leq 1$ we conclude the mapping is a contraction with modulus $\xi = (1-\lambda) + \lambda\gamma$. □



(a) (b) (c)

Figure 3: (Left) Learner (policy) / meta-learner (Q-fn): evolution of learning dynamics. Each sequence generates optimistic predictions for the other sequence to bootstrap on. **(Center)** Contours of the function $\xi(\gamma, \lambda) = (1-\lambda) + \lambda\gamma$ for $\lambda \in [0,1], \gamma \in [0,1]$. x-axis: λ, y-axis:γ. **(Right)**. Illustration of convergence dynamics (decrease of some error $\epsilon_0 = 1$, under some common choices for γ and λ. x-axis: iteration t, y-axis: error ϵ.

Since $\gamma \leq \xi$, as $\lambda \to 1$, we recover the standard Bellman operator with contraction rate γ (cf. Fig. 3)

Lemma 2. ***(Monotonicity)*** *The mapping $\mathcal{P}^{\lambda}_{\pi}$ in Def.2 is monotone.*

Proof. Monotonicity means

$$Q \leq Q' \implies \mathcal{P}^{\lambda}_{\pi} Q \leq \mathcal{P}^{\lambda}_{\pi} Q' \tag{60}$$

Using Asm. 2, and the convexity of the policy class for $\pi' = (1-\kappa)\pi + \kappa\pi^+$. The class of linear policies is convex. That is, the policy $\pi' = (1-\kappa)\pi + \kappa\pi^+$ is a linear policy for any given linear policies π, π^+ and some $\kappa \in [0,1]$. This implies $\mathcal{T}_{\pi'} Q \leq \mathcal{T}_{\pi'} Q'$, and the conclusion follows. □

B.3.2 Convergence

Consider first the simple setting in which the hints are one-step evaluations $Q^+_{t+1} = \mathcal{T}_{\pi_{t+1}} Q_t$.

Lemma 3. ***(Evaluation with $\mathcal{P}^{\lambda}_{\pi,\tilde{\pi}}$)*** *Consider the evaluation algorithm $Q_t = \mathcal{P}^{\lambda}_{\pi,\tilde{\pi}} Q_t$. Assume a behaviour policy π, and a target $\tilde{\pi}$, close in the sense $\max_s \|\pi(\cdot|s) - \tilde{\pi}(\cdot|s)\|_1 \leq \epsilon$. Then, the sequence $\{Q_t\}_{t \geq 0}\}$ converges geometrically to $Q_{\tilde{\pi}}$ at the rate $\mathcal{O}(\xi^t)$ indicated by the contraction factor $\xi = (1-\lambda) + \gamma\lambda\left((1-\kappa)\epsilon + \kappa\right) \leq 1$*

Proof. From Eq.50

$$\mathcal{P}^{\lambda}_{\pi,\tilde{\pi}} Q - Q_{\tilde{\pi}} = (1-\lambda)(Q - Q_{\tilde{\pi}}) + \lambda\left((1-\kappa)(\mathcal{T}_{\pi} Q - Q_{\tilde{\pi}}) + \kappa(\mathcal{T}_{\tilde{\pi}} Q - Q_{\tilde{\pi}})\right) \tag{61}$$

$$= \left((1-\lambda) + \lambda\gamma\left((1-\kappa)P_{\pi} + \kappa P_{\tilde{\pi}}\right)\right)(Q - Q_{\tilde{\pi}}) \tag{62}$$

Taking the max norm

$$\|\mathcal{P}^{\lambda}_{\pi,\tilde{\pi}} Q - Q_{\tilde{\pi}}\| \leq \xi \|Q - Q_{\tilde{\pi}}\| \tag{63}$$

for $\xi = (1-\lambda) + \gamma\lambda\left((1-\kappa)\epsilon + \kappa\right) \leq 1$, with ϵ s.t. π is ϵ-away from the target $\tilde{\pi}$

$$\|P_{\tilde{\pi}} - P_{\pi}\| = \sup_{\|Q\| \leq 1} \|(P_{\tilde{\pi}} - P_{\pi})Q\| \tag{64}$$

$$= \sup_{\|Q\| \leq 1} \max_{s,a} \left| \sum_{s'} P(s'|s,a) \sum_{a'} (\tilde{\pi}(a'|s') - \pi(a'|s')) Q(s',a') \right| \tag{65}$$

$$\leq \max_{s,a} \sum_{s'} P(s'|s,a) \sum_{a'} |\tilde{\pi}(a'|s') - \pi(a'|s')| \leq \epsilon \tag{66}$$

Combining this with Banach's fixed point theorem [Puterman, 1994], it is proven that $\mathcal{P}^{\lambda}_{\pi,\tilde{\pi}}$ with a fixed behavior policy π converges to a unique fixed point $Q_{\tilde{\pi}}$ at the rate $\mathcal{O}(\xi)$. □

Lemma 4. ***(Control with $\mathcal{P}^{\lambda}_{\pi,\tilde{\pi}}$)*** *Consider the control algorithm $Q_t = \mathcal{P}^{\lambda}_{\pi,\pi^+} Q_t$, in which the target at iteration t: $\pi_{t+1} = (1-\kappa)\pi_t + \kappa\pi^+_{t+1}$, is π^+_{t+1} greedy w.r.t. to the value estimate Q_t, $\pi^+_{t+1} = \arg\max_{\pi\in\Pi}\langle \pi, Q_\pi\rangle_d$.*

Then, the sequence $\{Q_t\}_{t\geq 0}\}$ converges geometrically to Q_π at the rate $\mathcal{O}(\xi^t)$ indicated by the contraction factor $\xi = (1-\lambda) + \gamma\lambda \leq 1$

Proof. From Eq.53

$$\mathcal{P}^{\lambda}_{\pi,\pi^+} Q - Q^* = (1-\lambda)(Q - Q^*) + \lambda\left((1-\kappa)(\mathcal{T}_\pi Q - Q^*) + \kappa(\mathcal{T}Q - Q^*)\right) \tag{67}$$

$$= ((1-\lambda) + \lambda\gamma\left((1-\kappa)P_\pi + \kappa P_{\pi^+}\right))(Q - Q^*) \tag{68}$$

Taking the max norm

$$\|\mathcal{P}^{\lambda}_{\pi,\pi^+} Q - Q^*\| \leq \xi\|Q - Q^*\| \tag{69}$$

for $\xi = (1-\lambda) + \gamma\lambda \leq 1$ □

For $h \geq 1$, cf. [Scherrer et al., 2015], there is no norm for which the operator $(\mathcal{P}^{\lambda}_{\pi})^h$ is a contraction.

Next, we use a general formulation for the propagation of errors through the iterations of the AOPI in Def. 1, that relies on the contraction and monotonicity properties from Sec. B.3.1, together with the progress bounds from Sec. B.3.4 (restated and adapted from Bertsekas [2012]) to conclude the rate of convergence.

The next theorem follows the same steps of convergence of optimistic PI from Bertsekas [2012], and proves the convergence of the AOPI algorithm in Def.1, by viewing it as a generalized instance of optimistic policy iteration.

Theorem 1. ***(Convergence of AOPI)*** *Let $\{\pi_t, Q^+_t, Q_t\}_{t\geq 0}$ be the sequence generated by the optimistic PI algorithm in Def. 1, and let $Q_0 \leq \mathcal{P}^{\lambda}_{\pi_1} Q_0 + \epsilon$. Then the algorithm converges geometrically (linearly) $\lim_{t\to\infty}\|Q^* - Q_t\| = 0$, at the rate indicated by the contraction factor $\mathcal{O}(\xi^t)$, where $\xi = 1 - \lambda + \lambda\gamma$, to the fixed point of the operator $\mathcal{P}^{\lambda}_{\pi}$, i.e. Q^*. Convergence follows from the error bound*

$$Q_t - \frac{\xi^t}{1-\xi}\epsilon \leq Q^* \leq Q_t + \frac{(t+1)\xi^t}{1-\xi}\epsilon \tag{70}$$

Proof. Using $Q_0 \leq \mathcal{P}^{\lambda}_{\pi_1} Q_0 + \epsilon$ from the premise and using $\mathcal{H} \equiv \mathcal{P}^{\lambda}_{\pi}$ in Lemma 10, we have $Q_t \leq \mathcal{P}^{\lambda}_{\pi_{t+1}} Q_t + \xi^t\epsilon$. Using this together with Lemma 8(ii) with $t = 0$ and $\mathcal{P}^{\lambda}_{\pi}$ in place of $\mathcal{H}$ shows the left-hand side of Eq. 70

$$Q_t \leq Q^* + \frac{\xi^t}{1-\xi}\epsilon \tag{71}$$

Using $Q_0 \leq \mathcal{P}^{\lambda}_{\pi_1} Q_0 + \epsilon$ and the second inequality from Lemma 8(ii) with $\mathcal{P}^{\lambda}_{\pi}$ in place of $\mathcal{H}$

$$Q^* \leq (\mathcal{P}^{\lambda}_{\pi})^t Q_0 + \frac{\xi^t}{1-\xi}\epsilon, \text{ for } t = 0, 1\ldots \tag{72}$$

Using again $Q_0 \leq \mathcal{P}^{\lambda}_{\pi_1} Q_0 + \epsilon$ and Lemma 10

$$\mathcal{P}^{\lambda}_{\pi_{j+1}} Q_j \leq Q_{j+1} + \frac{\xi}{1-\xi}\epsilon, \text{ for all } j = 0\ldots t-1 \tag{73}$$

Applying $(\mathcal{P}^{\lambda}_{\pi})^{t-j-1}$, on both sides and using monotonicity and contraction of $\mathcal{P}^{\lambda}_{\pi}$, we obtain

$$(\mathcal{P}^{\lambda}_{\pi})^{t-j} Q_j \leq (\mathcal{P}^{\lambda}_{\pi})^{t-j-1} Q_{j+1} + \frac{\xi^t}{1-\xi}\epsilon \text{ for all } j = 0\ldots t-1 \tag{74}$$

Adding over $j = 0\ldots t-1$

$$(\mathcal{P}^{\lambda}_{\pi})^t Q_0 \leq Q_t + \sum_{j=0}^{t-1}\frac{\xi^{t-j}}{1-\xi}\xi^j\epsilon = Q_t + \sum_{j=0}^{t-1}\frac{\xi^t}{1-\xi}\epsilon = Q_t + \frac{t\xi^t}{1-\xi}\epsilon \tag{75}$$

Adding this to Eq. 72 yields the right-hand side of Eq. 70. From the error bound in Eq. 70, taking the limit yields $\lim_{t\to\infty}\|Q^* - Q_t\| = 0$. Finally, if the number of policies is finite, Eq. 171 implies that there can be only a finite number of iterations for which $\mathcal{P}^{\lambda}_{\pi_{t+1}} Q_t - \frac{\xi^t}{1-\xi}\epsilon \leq Q_{t+1} \leq \mathcal{P}^{\lambda}_{\pi_{t+2}} Q_{t+1} + \xi^{t+1}\epsilon$, so we must have equality for some t, at which time $\mathcal{P}^{\lambda}_{\pi_{t+1}} Q_t = Q^*$. □

B.3.3 Error propagation

We now consider error bounds for the case where the policy evaluation and policy improvement operations are approximate. We consider a variation of the algorithm in Def. 1 that generates a sequence of policies π_{t+1} and a corresponding sequence of approximate Q-fns:

Definition 3. (Approximate Adaptive Optimistic Policy Iteration (AAOPI)) Let $\{\pi_t, Q_t\}_{t\geq 0}$, be the policy and Q-fn. sequences, $\mathcal{P}^\lambda$ a proximal operator, $\mathcal{G}(Q)$ the greedy set of Q. The Q-fn sequence adapts using "hints" $\{Q_t^+\}_{t\geq 0}$—partial policy evaluations $Q_{t+1}^+ \equiv \mathcal{T}_{\pi_{t+1}} Q_t$. The two sequences evolve as

$$\pi_{t+1} \in \mathcal{G}^\kappa_{\pi_t}(Q_t) \equiv \{\pi | \pi = (1-\kappa)\pi_t + \kappa \pi^+_{t+1}\} \quad \pi^+_{t+1} \in \mathcal{G}_{\epsilon'_{t+1}}(Q_t) \equiv \{\pi : \mathcal{T}_\pi Q_t \geq \mathcal{T} Q - \epsilon'_{t+1}\} \tag{76}$$

$$Q_{t+1} = \mathcal{P}^\lambda_{\pi_{t+1}} Q_t \equiv (1-\lambda) Q_t + \lambda Q^+_{t+1} \qquad Q^+_{t+1} \equiv \mathcal{T}_{\pi_{t+1}} Q_t + \epsilon_{t+1} \tag{77}$$

The main strategy is to decompose the sub-optimality $V^* - V_{\pi_t}$ into two components $V^* - \langle \pi_t, Q_t - \lambda\epsilon_t\rangle$ and $\langle \pi_t, Q_t - \lambda\epsilon_t\rangle - V_{\pi_t}$, and then to upper bound each of them.

After t steps, we decompose

$$\begin{aligned}
V^* - V_{\pi_t} &= \langle \pi^*, Q^*\rangle - \langle \pi_t, \mathcal{P}^\lambda_{\pi_t} Q_{t-1} - \lambda\epsilon_t\rangle + \langle \pi_t, \mathcal{P}^\lambda_{\pi_t} Q_{t-1} - \lambda\epsilon_t\rangle - \langle \pi_t, Q_{\pi_t}\rangle && (78)\\
&\leq \langle \pi^*, \underbrace{Q^* - (\mathcal{P}^\lambda_{\pi_t} Q_{t-1} - \lambda\epsilon_t)}_{d_t}\rangle + \langle \pi_t, \underbrace{\mathcal{P}^\lambda_{\pi_t} Q_{t-1} - \lambda\epsilon_t - Q_{\pi_t}}_{s_t}\rangle && (79)\\
&= \langle \pi^*, d_t\rangle + \langle \pi_t, s_t\rangle && (80)
\end{aligned}$$

We use the following notation

$$\begin{aligned}
b_t &= Q_t - \mathcal{T}_{\pi_{t+1}} Q_t && \text{(Bellman residual)} && (81)\\
d_t &= Q^* - (Q_t - \lambda\epsilon_t) && \text{(decision error)} && (82)\\
s_t &= (Q_t - \lambda\epsilon_t) - Q_{\pi_t} && \text{(prediction error)} && (83)
\end{aligned}$$

The terms d_t and s_t can be expressed using b_t.

We first prove an upper bound for b_t.

Lemma 5. ***(Upper bound on*** b_t***)*** *Consider the approximate AAOPI in Def. 3. Let* $\mathbf{A}_t \equiv ((1-\lambda) + \lambda\gamma P_{\pi_t})$. *For any non-negative integer t, the following holds:*

$$b_t \leq \left(\prod_{k=0}^{t} \mathbf{A}_k\right) b_0 + \sum_{n=0}^{t} \left(\prod_{k=n+1}^{t} \mathbf{A}_k\right) (\lambda(I - \gamma P_{\pi_n})\epsilon_n + \kappa\epsilon'_n) \tag{84}$$

where $\mathbf{A}_0 = I$

Proof. First, since $\pi^+_{t+1} \in \mathcal{G}_{\epsilon'_{t+1}}(Q_t)$

$$\begin{aligned}
\mathcal{T}_{\pi_{t+1}} Q_t &= (1-\kappa)\mathcal{T}_{\pi_t} Q_t + \kappa \mathcal{T}_{\pi^+_{t+1}} Q_t && (85)\\
&\geq (1-\kappa)\mathcal{T}_{\pi_t} Q_t + \kappa \mathcal{T} Q_t - \kappa\epsilon'_{t+1} && (86)\\
&\geq \mathcal{T}_{\pi_t} Q_t - \kappa\epsilon'_{t+1} && (87)
\end{aligned}$$

Consequently

$$
\begin{aligned}
b_t &= Q_t - \mathcal{T}_{\pi_{t+1}} Q_t && (88)\\
&\leq Q_t - \mathcal{T}_{\pi_t} Q_t + \kappa \epsilon'_{t+1} && (89)\\
&\text{(adding and subtracting } \lambda(I - \gamma P_{\pi_t})\epsilon_t) && (90)\\
&= Q_t - \lambda(I - \gamma P_{\pi_t})\epsilon_t - \mathcal{T}_{\pi_t} Q_t + \lambda(I - \gamma P_{\pi_t})\epsilon_t + \kappa \epsilon'_{t+1} && (91)\\
&= Q_t - \lambda \epsilon_t - \mathcal{T}_{\pi_t} Q_t + \lambda \gamma P_{\pi_t} \epsilon_t + \lambda(I - \gamma P_{\pi_t})\epsilon_t + \kappa \epsilon'_{t+1} && (92)\\
&= Q_t - \lambda \epsilon_t - \mathcal{T}_{\pi_t}(Q_t - \lambda \epsilon_t) + \lambda(I - \gamma P_{\pi_t})\epsilon_t + \kappa \epsilon'_{t+1} && (93)\\
&\text{(by def. } Q_t = (1-\lambda)Q_{t-1} + \lambda \mathcal{T}_{\pi_t} Q_{t-1} + \lambda \epsilon_t) && (94)\\
&= (1-\lambda)Q_{t-1} + \lambda \mathcal{T}_{\pi_t} Q_{t-1} - \mathcal{T}_{\pi_t}\left((1-\lambda)Q_{t-1} + \lambda \mathcal{T}_{\pi_t} Q_{t-1}\right) + \lambda(I - \gamma P_{\pi_t})\epsilon_t + \kappa \epsilon'_{t+1} && (95)\\
&\text{(by linearity of } \mathcal{T}_{\pi_t}, \mathcal{T}_{\pi_t}\left((1-\lambda)Q_{t-1} + \lambda \mathcal{T}_{\pi_t} Q_{t-1}\right) = (1-\lambda)\mathcal{T}_{\pi_t} Q_{t-1} + \lambda \mathcal{T}_{\pi_t}\mathcal{T}_{\pi_t} Q_{t-1}) && (96)\\
&= (1-\lambda)Q_{t-1} + \lambda \mathcal{T}_{\pi_t} Q_{t-1} - (1-\lambda)\mathcal{T}_{\pi_t} Q_{t-1} + \lambda \mathcal{T}_{\pi_t}\mathcal{T}_{\pi_t} Q_{t-1} + \lambda(I - \gamma P_{\pi_t})\epsilon_t + \kappa \epsilon'_{t+1} && (97)\\
&\text{(re-grouping terms)} && (98)\\
&= (1-\lambda)(\underbrace{Q_{t-1} - \mathcal{T}_{\pi_t} Q_{t-1}}_{b_{t-1}}) + \lambda\left(\mathcal{T}_{\pi_t} Q_{t-1} + \mathcal{T}_{\pi_t}\mathcal{T}_{\pi_t} Q_{t-1}\right) + \lambda(I - \gamma P_{\pi_t})\epsilon_t + \kappa \epsilon'_{t+1} && (99)\\
&= \underbrace{((1-\lambda) + \lambda \gamma P_{\pi_t})}_{\mathbf{A}_t}(\underbrace{Q_{t-1} - \mathcal{T}_{\pi_t} Q_{t-1}}_{b_{t-1}}) + \lambda(I - \gamma P_{\pi_t})\epsilon_t + \kappa \epsilon'_{t+1} && (100)
\end{aligned}
$$

By induction on t

$$
b_t \leq \left(\prod_{k=0}^{t} \mathbf{A}_k\right) b_0 + \sum_{n=0}^{t} \left(\prod_{k=n+1}^{t} \mathbf{A}_k\right) \left(\lambda(I - \gamma P_{\pi_n})\epsilon_n + \kappa \epsilon'_{n+1}\right) \tag{101}
$$

□

Corollary 1. ***(Upper bound on*** b_t***)*** *Consider the approximate AAOPI in Def. 3. For any non-negative integer t, the following holds:*

$$
b_t \leq \xi^t \|b_0\| \mathbf{1} + \sum_{k=0}^{t-1} \xi^{t-k-1} (\lambda(1-\gamma)\|\epsilon_k\| \mathbf{1} + \kappa \|\epsilon'_{k+1}\| \mathbf{1}) = \bar{b}_t \tag{102}
$$

Proof. The conclusion follows from the fact that the iteration matrices $\{\mathbf{A}_t\}_{t \geq 0}$ are monotone contractions with modulus $\xi = 1 - \lambda + \lambda\gamma$. □

Next, we prove an upper bound for d_t.

Lemma 6. ***(Upper bound on*** d_t***)*** *For any non-negative integer t, the following holds*

$$
d_t \leq \nu^t d_0 + \lambda \sum_{k=0}^{t-1} \nu^{t-k-1} ((1-\kappa)\bar{b}_k - \nu \|\epsilon_k\| \mathbf{1} + \kappa \|\epsilon'_{k+1}\| \mathbf{1}) \tag{103}
$$

where $\nu \equiv (1 - \lambda\kappa) + \lambda\kappa\gamma$ *and* $\bar{b}_t$ *as in Corollary 1.*

Proof.

$$
\begin{aligned}
d_{t+1} &= Q^* - (Q_{t+1} - \lambda\epsilon_{t+1}) && (104)\\
&= Q^* - (\mathcal{P}^{\lambda}_{\pi_{t+1}} Q_t - \lambda\epsilon_{t+1})) && (105)\\
&\text{(by def. } \mathcal{P}^{\lambda}_{\pi_{t+1}} Q_t = (1-\lambda)Q_t + \lambda\mathcal{T}_{\pi_{t+1}} Q_t + \lambda\epsilon_{t+1}))\\
&= Q^* - (1-\lambda)Q_t - \lambda\mathcal{T}_{\pi_{t+1}} Q_t && (106)\\
&\text{(by linearity and def. of } \pi_{t+1}, \mathcal{T}_{\pi_{t+1}} Q_t \geq (1-\kappa)\mathcal{T}_{\pi_t} Q_t + \kappa\mathcal{T} Q_t - \kappa\epsilon'_{t+1})\\
&\leq Q^* - (1-\lambda)Q_t - \lambda((1-\kappa)\mathcal{T}_{\pi_t} Q_t + \kappa\mathcal{T} Q_t) + \lambda\kappa\epsilon'_{t+1} && (107)\\
&\text{(since } (\mathcal{T}Q^* = (\lambda\kappa + (1-\lambda\kappa))\mathcal{T}Q^*)\\
&\leq (\lambda\kappa + (1-\lambda\kappa))\mathcal{T}Q^* - (1-\lambda)Q_t - \lambda((1-\kappa)\mathcal{T}_{\pi_t} Q_t + \kappa\mathcal{T} Q_t) + \lambda\kappa\epsilon'_{t+1} && (108)\\
&\text{(distributing } \mathcal{T}Q^*)\\
&\leq \lambda\kappa(\mathcal{T}Q^* - \mathcal{T}Q_t) + (1-\lambda\kappa)(Q^* - Q_t) + \lambda(1-\kappa)(Q_t - \mathcal{T}_{\pi_t} Q_t) + \lambda\kappa\epsilon'_{t+1} && (109)\\
&\leq \lambda\kappa\gamma P_{\pi^*}(Q^* - Q_t) + (1-\lambda\kappa)(Q^* - Q_t) + \lambda(1-\kappa)(Q_t - \mathcal{T}_{\pi_t} Q_t) + \lambda\kappa\epsilon'_{t+1} && (110)\\
&\leq ((1-\lambda\kappa) + \lambda\kappa\gamma P_{\pi^*})(Q^* - Q_t) + \lambda(1-\kappa)\underbrace{(Q_t - \mathcal{T}_{\pi_t} Q_t)}_{b_t} + \lambda\kappa\epsilon'_{t+1} && (111)\\
&\text{(adding and subtracting } \lambda((1-\lambda\kappa) + \lambda\kappa\gamma P_{\pi^*})\epsilon_t)\\
&\leq ((1-\lambda\kappa) + \lambda\kappa\gamma P_{\pi^*})\underbrace{(Q^* - (Q_t - \lambda\epsilon_t))}_{d_t} - \lambda((1-\lambda\kappa) + \lambda\kappa\gamma P_{\pi^*})\epsilon_t\\
&\quad + \lambda(1-\kappa)\underbrace{(Q_t - \mathcal{T}_{\pi_t} Q_t)}_{b_t} + \lambda\kappa\epsilon'_{t+1} && (112)
\end{aligned}
$$

Let $\nu \equiv (1-\lambda\kappa) + \lambda\kappa\gamma$,

$$
d_{t+1} \leq ((1-\lambda\kappa) + \lambda\kappa\gamma P_{\pi^*})d_t - \lambda\nu\|\epsilon_t\|\mathbf{1} + \lambda(1-\kappa)\bar{b}_t + \lambda\kappa\|\epsilon'_{t+1}\|\mathbf{1} \tag{113}
$$

By induction on t

$$
Q^* - Q_t \leq ((1-\lambda\kappa) + \lambda\kappa\gamma P_{\pi^*})^t (Q^* - Q_0) + \lambda\sum_{k=0}^{t-1} \nu^{t-k-1}((1-\kappa)\bar{b}_k - \nu\|\epsilon_k\|\mathbf{1} + \kappa\|\epsilon'_{k+1}\|\mathbf{1}) \tag{114}
$$

$$
\leq \nu^t\|Q^* - Q_0\|\mathbf{1} + \lambda\sum_{k=0}^{t-1} \nu^{t-k-1}((1-\kappa)\bar{b}_k - \nu\|\epsilon_k\|\mathbf{1} + \kappa\|\epsilon'_{k+1}\|\mathbf{1}) \tag{115}
$$

□

Lemma 7. ***(Upper bound on*** s_t***)*** *For any non-negative integer t, the following holds*

$$
s_t \leq \xi(1-\gamma)^{-1}\bar{b}_{t-1} \tag{116}
$$

for $\xi \equiv (1-\lambda) + \lambda\gamma$*, and* $\bar{b}$ *as in Corollary 1.*

Proof. From the definition

$$
\begin{aligned}
s_t &= (Q_t - \lambda\epsilon_t) - Q_{\pi_t} && (117)\\
&= (\mathcal{P}^{\lambda}_{\pi_t} Q_{t-1} - \lambda\epsilon_t) - Q_{\pi_t} && (118)\\
&= \mathcal{P}^{\lambda}_{\pi_t} Q_{t-1} - \lambda\epsilon_t - \mathcal{T}^{\infty}_{\pi_t} Q_{t-1} && (119)\\
&\text{(by def. } \mathcal{P}^{\lambda}_{\pi_t} Q_{t-1} = (1-\lambda)Q_{t-1} + \lambda\mathcal{T}_{\pi_t} Q_{t-1} + \lambda\epsilon_t)) \\
&= (1-\lambda)Q_{t-1} + \lambda\mathcal{T}_{\pi_t} Q_{t-1} - \mathcal{T}^{\infty}_{\pi_t} Q_{t-1} && (120)\\
&= (1-\lambda)(Q_{t-1} - \mathcal{T}^{\infty}_{\pi_t} Q_{t-1}) + \lambda(\mathcal{T}_{\pi_t} Q_{t-1} - \mathcal{T}^{\infty}_{\pi_t} Q_{t-1}) && (121)\\
&= (1-\lambda)(Q_{t-1} - \mathcal{T}^{\infty}_{\pi_t} Q_{t-1}) + \lambda\gamma P_{\pi_t}(Q_{t-1} - \mathcal{T}^{\infty}_{\pi_t} Q_{t-1}) && (122)\\
&= ((1-\lambda) + \lambda\gamma P_{\pi_t})(Q_{t-1} - \mathcal{T}^{\infty}_{\pi_t} Q_{t-1}) && (123)\\
&= ((1-\lambda) + \lambda\gamma P_{\pi_t})(Q_{t-1} - \mathcal{T}_{\pi_t} Q_{t-1} + \mathcal{T}_{\pi_t} Q_{t-1} - \mathcal{T}^2_{\pi_t} Q_{t-1} + \mathcal{T}^2_{\pi_t} Q_{t-1} \dots) && (124)\\
&= ((1-\lambda) + \lambda\gamma P_{\pi_t})((Q_{t-1} - \mathcal{T}_{\pi_t} Q_{t-1}) + \gamma P_{\pi_t}(Q_{t-1} - \mathcal{T}_{\pi_t} Q_{t-1}) + \gamma^2 P^2_{\pi_t}(Q_{t-1} - \mathcal{T}_{\pi_t} Q_{t-1} \dots) && (125)\\
&= ((1-\lambda) + \lambda\gamma P_{\pi_t}) \sum_{k=0}^{\infty} P^k_{\pi_t}(Q_{t-1} - \mathcal{T}_{\pi_t} Q_{t-1}) && (126)\\
&\text{(von Neumman series)} && (127)\\
&= ((1-\lambda) + \lambda\gamma P_{\pi_t})(I - \gamma P_{\pi_t})^{-1}(\underbrace{Q_{t-1} - \mathcal{T}_{\pi_t} Q_{t-1}}_{b_{t-1}}) && (128)
\end{aligned}
$$

Concluding with

$$
s_t = ((1-\lambda) + \lambda\gamma P_{\pi_t})(I - \gamma P_{\pi_t})^{-1} b_{t-1} \tag{129}
$$

Taking the max norm and upper bounding gives the result. □

Theorem 2. ***(Error propagation in AAOPI)*** *The following error bound reauthorizes the convergence of approximate AAOPI in Def. 3*

$$
V^* - V_{\pi_t} \leq \nu^t \|Q^* - Q_0\| + \frac{\nu^t}{(1-\gamma)} \|b_0\|_{\mathbf{1}} + \sum_{k=0}^{t-1} \nu^{t-k-1} \left(\lambda \|\epsilon_k\| \mathbf{1} + \frac{(1+\lambda(1-\gamma))\kappa}{(1-\gamma)} \|\epsilon'_{k+1}\| \mathbf{1} \right) \tag{130}
$$

for $\nu \equiv ((1-\lambda\kappa) + \kappa\lambda\gamma)$ *and* $\xi \equiv ((1-\lambda) + \lambda\gamma)$ *and* $b_0 = Q_0 - \mathcal{T}_{\pi_1} Q_0$.

Proof. Let $\xi \equiv ((1-\lambda) + \lambda\gamma)$. From Corollary 1 and Lemma 6

$$
\begin{aligned}
V^* - V_{\pi_t} &= \langle \pi^*, Q^* \rangle - \langle \pi_t, \mathcal{P}^{\lambda}_{\pi_t} Q_{t-1} \rangle + \langle \pi_t, \mathcal{P}^{\lambda}_{\pi_t} Q_{t-1} \rangle - \langle \pi_t, Q_{\pi_t} \rangle && (131)\\
&\leq \langle \pi^*, \underbrace{Q^* - (\mathcal{P}^{\lambda}_{\pi_t} Q_{t-1} - \lambda\epsilon_t)}_{d_t} \rangle + \langle \pi_t, \underbrace{\mathcal{P}^{\lambda}_{\pi_t} Q_{t-1} - \lambda\epsilon_t - Q_{\pi_t}}_{s_t} \rangle && (132)\\
&= \langle \pi^*, d_t \rangle + \langle \pi_t, s_t \rangle && (133)\\
&\leq \nu^t \|Q^* - Q_0\| + \lambda \left(\sum_{k=0}^{t-1} \nu^{t-k-1} \left((1-\kappa)\bar{b}_k - \nu\|\epsilon_k\|\mathbf{1} + \kappa\|\epsilon'_{k+1}\|\mathbf{1} \right) \right) + \xi(1-\gamma)^{-1}\bar{b}_t && (134)
\end{aligned}
$$

We simplify $\sum_{k=0}^{t-1} \nu^{t-1-k} \bar{b}_k$ as follows. Plugging in

$$
b_t \leq \xi^t \|b_0\| \mathbf{1} + \sum_{k=0}^{t-1} \xi^{t-k-1} (\lambda(1-\gamma)\|\epsilon_k\|\mathbf{1} + \kappa\|\epsilon'_{k+1}\|\mathbf{1}) = \bar{b}_t \tag{135}
$$

$$\sum_{k=0}^{t-1} \nu^{t-1-k} \bar{b}_k = \sum_{k=0}^{t-1} \nu^{t-1-k} \xi^k \|b_0\| \mathbf{1} + \sum_{k=0}^{t-1} \nu^{t-k-1} \sum_{l=0}^{k-1} \xi^{k-l-1} (\lambda(1-\gamma)\|\epsilon_l\| \mathbf{1} + \kappa \|\epsilon'_{l+1}\| \mathbf{1}) \tag{136}$$

$$= \nu^{t-1} \sum_{k=0}^{t-1} (\xi/\nu)^k \|b_0\| \mathbf{1} + \sum_{l=0}^{t-2} \sum_{k=l+1}^{t-1} \nu^{t-k-l} \xi^{k-l-1} (\lambda(1-\gamma)\|\epsilon_l\| \mathbf{1} + \kappa \|\epsilon'_{l+1}\| \mathbf{1}) \tag{137}$$

$$= \nu^{t-1} \frac{1-(\xi/\nu)^t}{1-(\xi/\nu)} \|b_0\| \mathbf{1} + \sum_{l=0}^{t-2} \sum_{k=0}^{t-l-2} \nu^{t-k-l-2} \xi^k (\lambda(1-\gamma)\|\epsilon_l\| \mathbf{1} + \kappa \|\epsilon'_{l+1}\| \mathbf{1}) \tag{138}$$

$$= \nu^t \frac{\nu^t - \xi^t}{\nu^t(\nu - \xi)} \|b_0\| \mathbf{1} + \sum_{l=0}^{t-2} \nu^{t-l-2} \sum_{k=0}^{t-l-2} (\xi/\nu)^k (\lambda(1-\gamma)\|\epsilon_l\| \mathbf{1} + \kappa \|\epsilon'_{l+1}\| \mathbf{1}) \tag{139}$$

$$= \nu^t \frac{\nu^t - \xi^t}{\nu^t(\nu - \xi)} \|b_0\| \mathbf{1} + \sum_{l=0}^{t-2} \nu^{t-l-2} \frac{1-(\xi/\nu)^{t-l-1}}{1-(\xi/\nu)} (\lambda(1-\gamma)\|\epsilon_l\| \mathbf{1} + \kappa \|\epsilon'_{l+1}\| \mathbf{1}) \tag{140}$$

$$= \frac{\nu^t - \xi^t}{\nu - \xi} \|b_0\| \mathbf{1} + \sum_{l=0}^{t-2} \frac{\nu^{t-l-1} - \xi^{t-l-1}}{\nu - \xi} (\lambda(1-\gamma)\|\epsilon_l\| \mathbf{1} + \kappa \|\epsilon'_{l+1}\| \mathbf{1}) \tag{141}$$

Using this result and

$$\nu - \xi = ((1-\lambda\kappa) + \kappa\lambda\gamma) - ((1-\lambda) + \lambda\gamma) = -\lambda\kappa + \kappa\lambda\gamma + \lambda - \lambda\gamma \tag{142}$$

$$= \lambda(1-\kappa) - (1-\kappa)\lambda\gamma = \lambda(1-\kappa)(1-\gamma) \tag{143}$$

we deduce that

$$V^* - V_{\pi_t} \leq \nu^t \|Q^* - Q_0\| + \frac{\nu^t - \xi^t}{(1-\gamma)} \|b_0\|_{\mathbf{1}} + \sum_{k=0}^{t-2} \frac{\nu^{t-k-1} - \xi^{t-k-1}}{(1-\gamma)} (\lambda(1-\gamma)\|\epsilon_k\|\mathbf{1} + \kappa\|\epsilon'_{k+1}\|\mathbf{1}) \tag{144}$$

$$+ \sum_{k=0}^{t-1} \nu^{t-k-1} \left((-\lambda\nu)\|\epsilon_k\|\mathbf{1} + \lambda\kappa\|\epsilon'_{k+1}\|\mathbf{1}\right) + \xi(1-\gamma)^{-1}\bar{b}_t \tag{145}$$

$$\leq \nu^t \|Q^* - Q_0\| + \frac{\nu^t - \xi^t}{(1-\gamma)} \|b_0\|_{\mathbf{1}} + \tag{146}$$

$$\sum_{k=0}^{t-2} (\nu^{t-k-1} - \xi^{t-k-1}) \left(\lambda|\epsilon_k\|\mathbf{1} + \frac{\kappa}{(1-\gamma)}\|\epsilon'_{k+1}\|\mathbf{1}\right) \tag{147}$$

$$+ \sum_{k=0}^{t-1} \nu^{t-k-1} \left((-\lambda\nu)\|\epsilon_k\|\mathbf{1} + \lambda\kappa\|\epsilon'_{k+1}\|\mathbf{1}\right) \tag{148}$$

$$+ \xi \frac{\xi^t}{(1-\gamma)} \|b_0\|\mathbf{1} + \sum_{k=0}^{t-1} \xi^{t-k} (\lambda\|\epsilon_k\|\mathbf{1} + \kappa \frac{1}{(1-\gamma)} \|\epsilon'_{k+1}\|\mathbf{1}) \tag{149}$$

$$\leq \nu^t \|Q^* - Q_0\| + \frac{\nu^t}{(1-\gamma)} \|b_0\|_{\mathbf{1}} + \lambda \sum_{k=0}^{t-2} (\nu^{t-k-1} - \xi^{t-k-1}) \|\epsilon_k\|\mathbf{1} \tag{150}$$

$$+ \kappa \sum_{k=0}^{t-2} (\nu^{t-k-1} - \xi^{t-k-1}) \frac{1}{(1-\gamma)} \|\epsilon'_{k+1}\|\mathbf{1}) \tag{151}$$

$$+ \sum_{k=0}^{t-1} \nu^{t-k-1} \left((-\lambda\nu)\|\epsilon_k\|\mathbf{1} + \lambda\kappa\|\epsilon'_{k+1}\|\mathbf{1}\right) \tag{152}$$

$$+ \sum_{k=0}^{t-1} \xi^{t-k} (\lambda\|\epsilon_k\|\mathbf{1} + \kappa \frac{1}{(1-\gamma)} \|\epsilon'_{k+1}\|\mathbf{1}) \tag{153}$$

$$\leq \nu^t \|Q^* - Q_0\| + \frac{\nu^t}{(1-\gamma)} \|b_0\|_{\mathbf{1}} + \lambda \sum_{k=0}^{t-1} (\nu^{t-k-1} - \xi^{t-k-1}) \|\epsilon_k\|\mathbf{1} \tag{154}$$

$$+ \kappa \sum_{k=0}^{t-1} (\nu^{t-k-1} - \xi^{t-k-1}) \frac{1}{(1-\gamma)} \|\epsilon'_{k+1}\|\mathbf{1} \tag{155}$$

$$+ \sum_{k=0}^{t-1} \nu^{t-k-1} \left((-\lambda\nu)\|\epsilon_k\|\mathbf{1} + \lambda\kappa\|\epsilon'_{k+1}\|\mathbf{1}\right) \tag{156}$$

$$+ \sum_{k=0}^{t-1} \xi^{t-k} (\lambda\|\epsilon_k\|\mathbf{1} + \kappa \frac{1}{(1-\gamma)} \|\epsilon'_{k+1}\|\mathbf{1}) \tag{157}$$

$$\leq \nu^t \|Q^* - Q_0\| + \frac{\nu^t}{(1-\gamma)} \|b_0\|_{\mathbf{1}} + \sum_{k=0}^{t-1} \nu^{t-k-1} \left(\lambda\|\epsilon_k\|\mathbf{1} + \frac{(1+\lambda(1-\gamma))\kappa}{(1-\gamma)} \|\epsilon'_{k+1}\|\mathbf{1}\right) \tag{158}$$

□

B.3.4 Helpful Lemmas

Lemma 8. ***(Error bounds: adapted from Lemma 6.1. [Bertsekas, 2012])*** *Let $\mathcal{H}$ be a mapping that satisfies the monotonicity assumption*

$$Q \leq Q' \implies \mathcal{H}Q \leq \mathcal{H}Q', \forall Q, Q' \tag{159}$$

and the contraction assumption

$$\|\mathcal{H}Q - \mathcal{H}Q'\| \leq \xi\|Q - Q'\|, \forall Q, Q' \tag{160}$$

for some $\xi \in (0,1)$.

(i) For all Q, Q' and error vector $\epsilon \succeq 0$, denoting per entry inequality, we have

$$Q \leq Q' + \epsilon \implies \mathcal{H}Q \leq \mathcal{H}Q' + \xi\epsilon \tag{161}$$

(ii) For all Q, $\epsilon \succeq 0$, and $t \in \mathbb{N}$, we have

$$Q \leq \mathcal{H}Q + \epsilon \implies \mathcal{H}^t Q \leq Q^* + \frac{\xi^t}{1-\xi}\epsilon \tag{162}$$

$$\mathcal{H}Q \leq Q + \epsilon \implies Q^* \leq \mathcal{H}^t Q + \frac{\xi^t}{1-\xi}\epsilon \tag{163}$$

where Q^* is the fixed point of $\mathcal{H}$.

Lemma 9. ***(Multi-step error bounds: adapted from Lemma 6.2. [Bertsekas, 2012])*** *Let $\mathcal{H}$ be a monotone contraction with modulus ξ.*

$$Q \leq \mathcal{H}Q + \epsilon \tag{164}$$

and let $\pi_{t+1} \in \mathcal{G}(\mathcal{H}Q_{t-1})$ Then for all $t \geq 0$, let $(\mathcal{H})^t$ denote the t-fold composition of the operators $\mathcal{H}$, in sequence. With this notation, we have

$$\mathcal{H}Q \leq (\mathcal{H})^t Q + \frac{\xi}{1-\xi}\epsilon \tag{165}$$

and

$$(\mathcal{H})^t Q \leq \mathcal{H}(\mathcal{H})^t Q + \xi^t \epsilon \tag{166}$$

Proof. Since

$$Q \leq \mathcal{H}Q + \epsilon \tag{167}$$

by Lemma 8(i) with $\mathcal{H}$ in place of the mapping $\mathcal{H}$ and $Q' = \mathcal{H}Q$, we have, $\forall j \geq 1$

$$(\mathcal{H})^j Q \leq (\mathcal{H})^{j+1} Q + \xi^j \epsilon \tag{168}$$

By adding this relation over $j = 1, \ldots, t-1$, we have

$$\mathcal{H}Q \leq (\mathcal{H})^t Q + \sum_{j=1}^{t-1} \xi^j \epsilon = (\mathcal{H})^t Q - \frac{\xi - \xi^t}{1-\xi}\epsilon \leq (\mathcal{H})^t Q + \frac{\xi}{1-\xi}\epsilon \tag{169}$$

showing Eq. 165. From Eq. 168, for $j = t$, we obtain Eq. 166

$$(\mathcal{H})^t Q \leq (\mathcal{H})^{t+1} Q + \xi^t \epsilon = \mathcal{H}((\mathcal{H})^t Q) + \xi^t \epsilon \tag{170}$$

□

Lemma 10. ***(Progress improvement error bound: adapted from Lemma 6.3. [Bertsekas, 2012])*** *Let $\mathcal{H}^\xi_{\pi_t}, t \geq 0$ be monotone contractions with modulus ξ, s.t. $Q_0 \leq \mathcal{H}^\xi_{\pi_1} Q_0 + \epsilon$, Then for all $t \geq 0$*

$$\mathcal{H}^\xi_{\pi_{t+1}} Q_t - \frac{\xi^t}{1-\xi}\epsilon \leq Q_{t+1} \leq \mathcal{H}^\xi_{\pi_{t+2}} Q_{t+1} + \xi^{t+1}\epsilon \tag{171}$$

Proof. By induction of t using Lemma 9. For the starting condition $t = 0$, using $Q = Q_0$, $\pi = \pi_0$ in Eq. 165

$$\mathcal{H}^\xi_{\pi_1} Q_0 \leq Q_1 + \frac{\xi}{1-\xi}\epsilon \tag{172}$$

and in Eq. 166

$$Q_1 \leq \mathcal{H}^\xi_{\pi_2} Q_1 + \frac{\xi}{1-\xi}\epsilon \tag{173}$$

Then, assuming Eq. 171 holds for all $t \geq 1$

$$Q_t \leq \mathcal{H}^\xi_{\pi_{t+1}} Q_t + \xi^t \epsilon \tag{174}$$

and using Eq. 165 and Eq. 166 with $Q = Q_t$, $\pi = \pi_t$, we obtain

$$\mathcal{H}^\xi_{\pi_{t+1}} Q_t \leq Q_{t+1} + \frac{\xi}{1-\xi}\epsilon \tag{175}$$

and

$$Q_{t+1} \leq \mathcal{H}^\xi_{\pi_{t+2}} Q_{t+1} + \xi^t \epsilon \tag{176}$$

which completes the induction. □

C Empirical analysis details

Table 4: Notation

t	iterations/timesteps
T	number of iterations
n	rollout length
$\mathcal{B}$	buffer
$\mathcal{M}$	meta-buffer
w	standard critic (Q-fn Q_w) parameters
η	meta parameters of meta-learner (g_η or Q_η)
ζ	step size for meta-learner's parameters η (Q-fn Q_η)
χ	step size for standard critic's parameters w (Q-fn Q_w)
β	step size for the policy learner's parameters θ (π_θ)
h	lookahead horizon
Q^h	search Q-values up to lookahead horizon h (tree depth)

C.1 Algorithms

Algorithm 2 **Policy gradient**

1: **Init:** params θ_0, buffer $\mathcal{B} = [()]$
2: **for** $t \in 0..T$ iterations **do**
3: Every n steps using a rollout $\mathcal{B} \leftarrow (S_t, A_t, R_t, S_{t+1} \dots S_{t+n}) \sim \pi_{\theta_t}$
4: Update policy learner $\pi_{\theta_{t+1}}$ cf. Eq.177
5: **end for**

Policy gradients Algorithm 2 describes a standard PG algorithm (cf. [Williams, 1992]) with an expert oracle critic Q_{π_θ}, for the policy evaluation of π_θ. The standard policy gradient update is

$$\theta_{t+1} = \theta_t + \zeta \frac{1}{n} \sum_{i=t}^{t+n} \nabla_{\theta_t} \log \pi_{\theta_t}(A_i|S_i) \left(Q_{\pi_{\theta_t}}(S_i, A_i) - \mathbb{E}_{\pi_{\theta_t}}[Q_{\pi_{\theta_t}}(S_i, \cdot)] \right) \tag{177}$$

Algorithm 3 **Actor-critic**

1: **Init:** params (θ_0, w_0), buffer $\mathcal{B} = [()]$
2: **for** $t \in 0..T$ iterations **do**
3: Every n steps using a rollout $\mathcal{B} \leftarrow (S_t, A_t, R_t, S_{t+1} \dots S_{t+n}) \sim \pi_{\theta_t}$
4: Update critic $Q_{w_{t+1}}$ cf. Eq.179 and policy learner $\pi_{\theta_{t+1}}$ cf. Eq.178
5: **end for**

Actor-critic Algorithm 3 describes a standard AC algorithm (cf. [Sutton et al., 1999]) with an estimated critic Q_w, for the policy evaluation of π_θ. The policy updates

$$\theta_{t+1} = \theta_t + \beta \frac{1}{n} \sum_{i=t}^{t+n} \nabla_{\theta_t} \log \pi_{\theta_t}(A_i|S_i) \left(Q_{w_t}(S_i, A_i) - \mathbb{E}_{\pi_{\theta_t}}[Q_{w_t}(S_i, \cdot)] \right) \tag{178}$$

and the critic's update using TD(0) learning, writes

$$w_{t+1} = w_t - \chi \frac{1}{n} \sum_{i=t}^{t+n} \left(R_i + \gamma E_{\pi_t}[Q_{w_t}(S_{i+1}, \cdot)] - Q_{w_t}(S_i, A_i) \right) \nabla_{w_t} Q_{w_t}(S_i, A_i) \tag{179}$$

Algorithm 4 **Policy gradients with forward search**

Init: params (θ_0, w_0), buffer $\mathcal{B} = [()]$
for $t \in 0..T$ iterations **do**
  Every n steps using a rollout $\mathcal{B} \leftarrow (S_t, A_t, R_t, S_{t+1} \dots S_{t+n}) \sim \pi_{\theta_t}$
    Generate search Q-values Q_t^h up to lookahead horizon h with
      (i) $Q_t^h = \mathcal{T}_{\pi_t}^h Q_{w_t}$
      (ii) $Q_t^h = \mathcal{T}^h Q_{w_t}$
    Update critic $Q_{w_{t+1}}$ cf. Eq.181 and policy learner $\pi_{\theta_{t+1}}$ cf. Eq.180, using Q_t^h
end for

Forward search with a model Algorithm 4 describes an AC algorithm with h-step lookahead search in the gradient critic

$$\theta_{t+1} = \theta_t + \beta \frac{1}{n} \sum_{i=0}^{n} \nabla_{\theta_t} \log \pi_{\theta_t}(A_i|S_i) \left(Q_t^h(S_i, A_i) - \mathbb{E}_{\pi_t}[Q_t^h(S_i, \cdot)] \right) \tag{180}$$

where Q_t^h is either (i) $Q_t^h = \mathcal{T}^h Q_{w_t}$ or (ii) $Q_t^h = \mathcal{T}^h Q_{w_t}$, depending on the experimental setup, and the critic is updated toward the search Q-values

$$w_{t+1} = w_t - \chi \frac{1}{n} \sum_{i=0}^{n} \nabla_{w_t} Q_{w_t}(S_i, A_i) \left(R_i + \gamma E_{\pi_t}[Q_t^h(S_{i+1}, \cdot)] - Q_{w_t}(S_i, A_i) \right) \tag{181}$$

Algorithm 5 **Optimistic policy gradients with policy targets computed from expert hints**

Init: params (θ_0, η_0), buffer $\mathcal{B} = [()]$, meta-buffer $\mathcal{M} = [\mathcal{B}, ..]$
for $t \in 0..T$ iterations **do**
  Every n steps using a rollout $\mathcal{B}_t \leftarrow (S_t, A_t, R_t, S_{t+1} \dots S_{t+n}) \sim \pi_{\theta_t}$
    Predict $g_{\eta_t} = \varphi(Q_{\eta_t})$
    Update learner with using optimistic prediction g_{η_t} using Eq. 182
    Every h steps using experience stored in the meta-buffer $\mathcal{M} \leftarrow (\mathcal{B}_t, \dots \mathcal{B}_{t+h})$
      Compute policy targets $\pi_{\tilde{\theta}_{t+1}}$ cf. Eq. 183 or $\tilde{\pi}_{t+1}$ cf. Eq. 184
      Update meta-learner $g_{\eta_{t+1}}$ cf. Eq. 185
end for

Optimistic policy gradients with expert targets (hints) Algorithm 5 describes a meta-gradient based algorithm for learning *optimistic policy gradients* by supervised learning from policy targets computed with complete hints (Q_π). The meta-update used updates

$$\theta_{t+1} = \theta_t + \beta g_{\eta_t} \tag{182}$$

where $g_{\eta_t} = \varphi_{\pi_{\theta_t}}(Q_{\eta_t}) = \frac{1}{n} \sum_{i=0}^{n} \nabla_{\theta_t} \log \pi_{\theta_t}(A_i|S_i) \left((Q_{\eta_t}(S_i, A_i) - \mathbb{E}_{\pi_{\theta_t}}[(Q_{\eta_t}(S_i, A_i)] \right)$ The policy targets are (i) *parametric policies* obtained at iteration t by starting from the parameters θ_{t+1} ($\tilde{\theta}_{t+1}^0 = \theta_{t+1}$) and executing h parameter updates with data from successive batches of rollouts $\mathcal{B}_{t+1:t+h}$ sampled from the meta-buffer $\mathcal{M}$

$$\tilde{\theta}_{t+1}^{j+1} = \tilde{\theta}_{t+1}^{j} + \beta g_j^+ \tag{183}$$

with hints $g_j^+ \equiv \varphi(Q_{\pi_{\tilde{\theta}_{t+1}^j}})$. After h steps the resulting target parameters $\tilde{\theta}_{t+1} \equiv \tilde{\theta}_{t+1}^h$, yield an h-step ahead target policy $\pi_{\tilde{\theta}_{t+1}}$.

The other choice we experiment with is to use a target constructed with (ii) *geometric updates* for one (or more) steps ahead, similarly to tree-search policy improvement procedures. The targets are initialized with $\pi_{t+1}^0 = \pi_{\theta_{t+1}}$ and execute one (or more) steps of policy improvement

$$\tilde{\pi}_{t+1}^{j+1} \propto \pi_{t+1}^{j} \exp \alpha Q_j^+ \tag{184}$$

with hints $Q_j^+ \equiv Q_{\pi_{t+1}^j}$, yielding the non-parametric policy target $\tilde{\pi}_{t+1} \equiv \tilde{\pi}_{t+1}^{j+1}$. Setting $\alpha \to \infty$ in Eq.184, if the hints are given, or can be computed with the help of the simulator model, we obtain an update similar to the multi-step greedy operator $\mathcal{T}^h$ used in forward search.

The next parameter vector η_{t+1} for the gradient g_η is distilled via meta-gradient learning by projecting the expert policy target $\tilde{\pi}_{t+1}$ (or $\pi_{\tilde{\theta}_{t+1}}$) using the data samples from $\mathcal{B}_t, \dots \mathcal{B}_{t+h}$ from $\mathcal{M}$ and the surrogate objective

$$\eta_{t+1} = \eta_t - \zeta \frac{1}{h} \sum_{j=t}^{t+h} \nabla_{\eta_t} \mathrm{KL}(\pi_{\theta_{t+1}}(S_j), \tilde{\pi}_{t+1}(S_j)) \tag{185}$$

Algorithm 6 Optimistic policy gradients with target predictions

Init: params (θ_0, w_0, η_0), buffer $\mathcal{B} = [()]$, meta-buffer $\mathcal{M} = [\mathcal{B}, ..]$
for $t \in 0..T$ iterations **do**
  Every n steps using a rollout $\mathcal{B}_t \leftarrow (S_t, A_t, R_t, S_{t+1} \dots S_{t+n}) \sim \pi_{\theta_t}$
    Predict $g_{\eta_t} = \varphi(Q_{\eta_t})$
    Update learner with optimistic prediction g_{η_t} using Eq. 182
    Update $Q_{w_{t+1}}$ cf. Eq.179
    Every h steps using experience stored in the meta-buffer $\mathcal{M} \leftarrow (\mathcal{B}_t, \dots \mathcal{B}_{t+h})$
      Compute policy targets $\pi_{\tilde{\theta}_{t+1}}$ cf. Eq. 186 or $\tilde{\pi}_{t+1}$ cf. Eq. 187
      Update meta-learner $g_{\eta_{t+1}}$ cf. Eq. 185
end for

Optimistic policy gradients with target predictions Algorithm 6 describes a meta-gradient based algorithm for learning *optimistic policy gradients*, by self-supervision from target predictions (learned estimators). The targets we use are (i) *parametric*, computed at iteration t, similarly to the previous paragraph (Eq. 183, except we now replace the hint that was using complete evaluations, with a partial evaluation

$$g_j^+ \equiv \varphi(Q_{w_{t+j}}) \tag{186}$$

We also experiment with the (ii) *non-parametric target* that takes *geometric* steps, similarly to tree-search policy improvement procedure, where the hint from Eq.184, using complete evaluations, is replaced with

$$Q_j^+ \equiv Q_{w_{t+j}} \tag{187}$$

C.2 Experimental setup

Environment details All empirical studies are performed on the same discrete navigation task from Sutton and Barto [2018], illustrated in Fig. 4. "G" marks the position of the goal and the end of an episode. "S" denotes the starting state to which the agent is reset at the end of the episode. The state space size is 48, $\gamma = 0.99$. There are 4 actions that can transition the agent to each one of the adjacent states. Reward is 1 at the goal, and zero everywhere else. Episodes terminate and restart from the initial state upon reaching the goal.

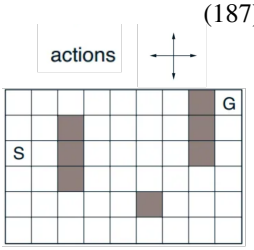


Figure 4: Maze Navigation: illustration of the MDP used in the empirical studies

Protocol All empirical studies report the regret of policy performance every step, and at every episode $V_\rho(\pi^*) - V(\pi_t)$, for a maximum number of 500 episodes. Hyperparameter sensitivity plots show the cumulative regret per total number of steps of experience cumulated in 500 episodes, $\sum_t V_\rho(\pi^*) - V(\pi_t)$. This quantity captures the sample efficiency in terms of number of steps of interaction required.

Algorithmic implementation details Meta-gradient based algorithms keep parametric representations of the gradient fields via a parametric advantage $\mathrm{A}_\eta(s,a) = Q_\eta(s,a) - V_\eta(s), \forall s, a$, s.t. a learned gradient update consists of a parametric gradient step on the loss

$$\theta_{t+1} = \theta_t + \nabla \mathcal{L}(\theta; \mathcal{B}_t)\Big|_{\theta=\theta_t} \tag{188}$$

$$\mathcal{L}(\theta; \mathcal{B}_t) = \frac{1}{n} \sum_{i=t}^{t+n} \log \pi_\theta(A_i|S_i) \left(Q_\eta(S_i, A_i) - \sum_a \pi_{\theta_t}(A_i|S_i) Q_\eta(S_i, A_i) \right) \tag{189}$$

Policies use the standard softmax transform $\pi_\theta = \frac{\exp f_\theta(s,a)}{\sum_b \exp f_\theta(s,b)}$, with f_θ, the policy logits. In the experiments illustrated we use a tabular, one-hot representation of the state space as features, so f_θ is essentially θ. The same holds for the critic's parameter vector Q_w, and the meta-learner's parameter vector Q_η.

The experiments were written using JAX, Haiku, and Optax [Bradbury et al., 2018, Babuschkin et al., 2020, Hennigan et al., 2020].

Experimental details for the forward search experiment We used forward search with the environment true dynamics model up to horizon h, backing-up the current value estimate Q_{w_t} at the leaves. We distinguish between two settings: (i) using the previous policy π_t for *bootstrapping* in the tree-search back-up procedure, i.e. obtaining $Q_t^h = \mathcal{T}_{\pi_t}^h Q_{w_t}$ at the root of the tree; or (ii) using the greedification inside the tree to obtain $Q_t^h = \mathcal{T}^h Q_{w_t}$ at the root. Table 5 specifies the hyperparameters used for both of the aforementioned experimental settings. Results shown in the main text are averaged over 10 seeds and show the standard error over runs.

Table 5: Hyperparameters for optimism via forward search on the Maze Gridworld in Fig. 4

Hyperparameter	
β (policy step size)	0.5
χ (Q-fn step size)	{0.01, 0.1, 0.5, 0.9}
h (lookahead horizon)	{0, 1, 2, 4, 8, 16}
n (rollout length)	2
policy/Q-fn optimiser	SGD

Experimental details for the meta-gradient experiments with expert target/hints For this experiment we used Algorithm 5 described in Sec. C.1, and the hyperaparameters in Table 6.

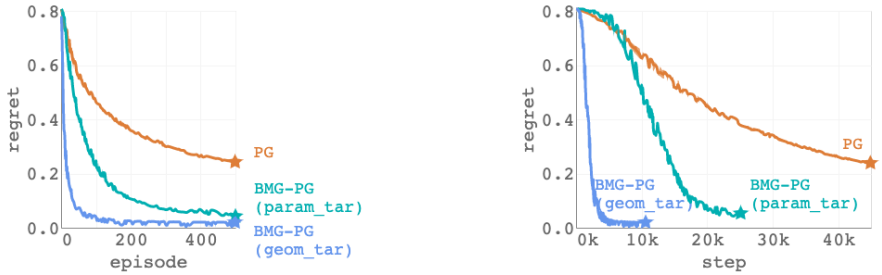
Table 6: Hyperparameters for optimism via meta-gradient learning with expert targets/hints on the Maze Gridworld in Fig. 4

Hyperparameter	
β (policy step size)	0.1 (training plots Fig.2-d) {0.1, 0.5} (sensitivity plots Fig.2-c)
χ (Q-fn step size)	-
h (lookahead horizon)	1
α (step size π)	1
n (rollout length)	2
policy optimiser	SGD
meta-learner optimiser	Adam

Experimental details for the meta-gradient experiments with target predictions/bootstrapping For this experiment we used Algorithm 6 described in Sec. C.1, and the hyperaparameters in Table 7.

Table 7: Hyperparameters for optimism via meta-gradient learning with target predictions/bootstrapping on the Maze Gridworld in Fig. 4

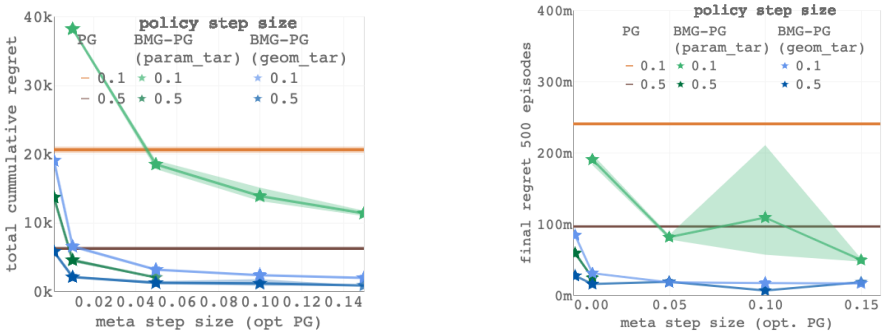
Hyperparameter	
β (policy step size)	0.5
χ (Q-fn step size)	0.1 (training plots Fig.2-e) {0.1, 0.5} (sensitivity plots Fig.2-f)
h (lookahead horizon)	1
n (rollout length)	2
policy optimiser	SGD
meta-learner optimiser	Adam



(a) expert targets, Adam, perf/episode

(b) expert targets, Adam, perf/step

Figure 5: Meta-learner uses Adam. Policy optimization with adaptive optimistic policy gradients. x-axis - (a) no of episodes, (b) no of steps. y-axis - regret $V_\rho(\pi^*) - V_\rho(\pi_t)$. Learning curves denote: the baseline - standard PG algorithm, *adaptive optimistic policy gradient learning algorithms* - with parametric target policies , functional non-parametric target policies, trained with meta-gradients from *expert targets*. Shades (wherever noticeable) denote standard error over different runs.



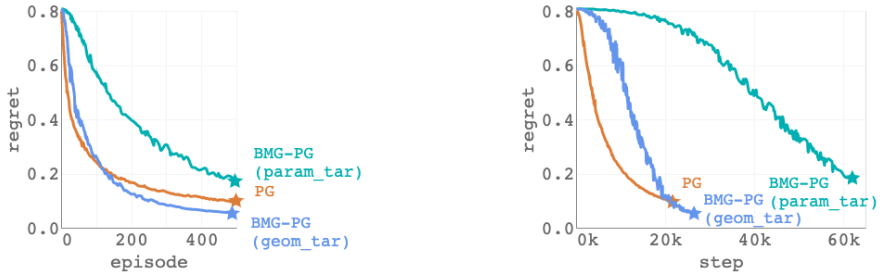
(a) Hyperparam. sensitivity / expert targets / Adam / total regret

(b) Hyperparam. sensitivity / expert targets / Adam / final regret

Figure 6: Meta-learner uses Adam. Hyper-parameter sensitivity curves for the meta-learning rate ζ - x-axis, y-axis - total cummulative regret (a): $\sum_{i \leq t} V_\rho(\pi^*) - V_\rho(\pi_i)$, final regret (b): $V_\rho(\pi^*) - V_\rho(\pi_T)$; Learning curves show *adaptive optimistic policy gradient learning algorithms* - with parametric target policies , functional non-parametric target policies, trained with meta-gradients from *expert targets*. Different tones show the evolution of the meta-hyperparameter ζ to those used in the inner learned optimization algorithm, i.e. the policy step size. Different straight lines denote the baseline—standard PG. Shades denote standard error over different runs.

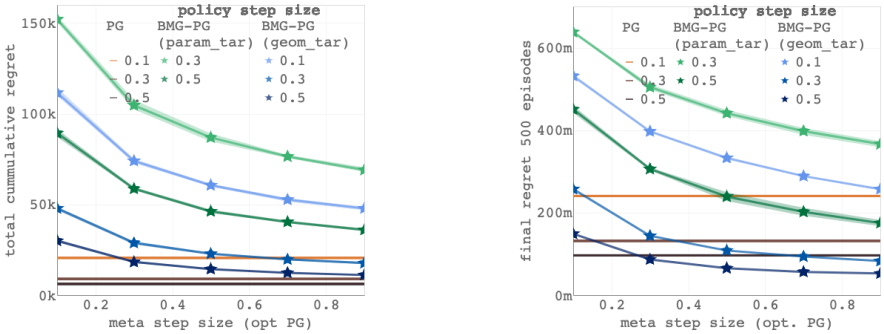
C.3 Additional results & observations

Fig 5 shows learning curves, and Fig 6 hyperparameter sensitivity, for experiments with expert targets, when using Adam for the meta-optimization. Fig 7 (learning curves), and Fig 8 (hyperparameter sensitivity) illustrate results for when the meta-optimization uses SGD. The next set of figures show experiments with target predictions—for Adam Fig 9 (learning curves) and Fig 10 (hyperparameter sensitivity), and for SGD Fig 11 (learning curves) and Fig 12 (hyperparameter sensitivity).



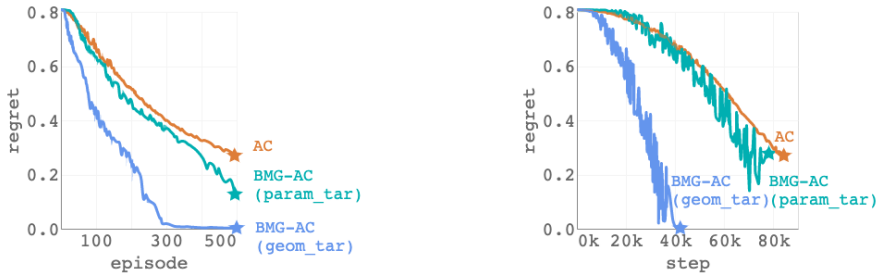
(a) expert targets, SGD, perf/episode (b) expert targets, SGD, perf/step

Figure 7: Meta-learner uses SGD. Policy optimization with adaptive optimistic policy gradients. x-axis - (a) no of episodes, (b) no of steps. y-axis - regret $V_\rho(\pi^*) - V_\rho(\pi_t)$. Learning curves denote: the baseline - standard PG algorithm, *optimistic policy gradient learning algorithms* - with parametric target policies , functional non-parametric target policies, trained with meta-gradients from *expert targets*. Shades (wherever noticeable) denote standard error over different runs.



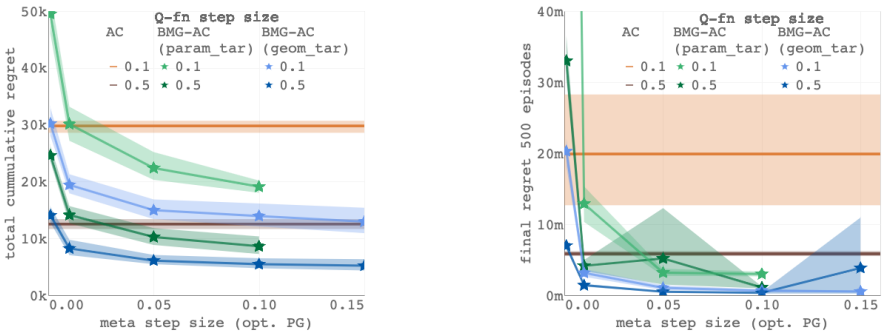
(a) expert targets, SGD, total regret (b) expert targets, SGD, final regret

Figure 8: Meta-learner uses SGD. Hyper-parameter sensitivity curves for the meta-learning rate ζ - x-axis. y-axis - total cumulative regret (a): $\sum_{i \leq t} V_\rho(\pi^*) - V_\rho(\pi_i)$, final regret (b): $V_\rho(\pi^*) - V_\rho(\pi_T)$. Learning curves show *adaptive optimistic policy gradient learning algorithms* - with parametric target policies, functional non-parametric target policies, trained with meta-gradients from *expert targets*. Different tones show the evolution of the meta-hyperparameter ζ to those used in the inner learned optimization algorithm, i.e. the policy step size. Different straight lines denote the baseline—standard PG. Shades denote standard error over different runs.



(a) target pred, Adam, perf/episode (b) target pred, Adam, perf/step

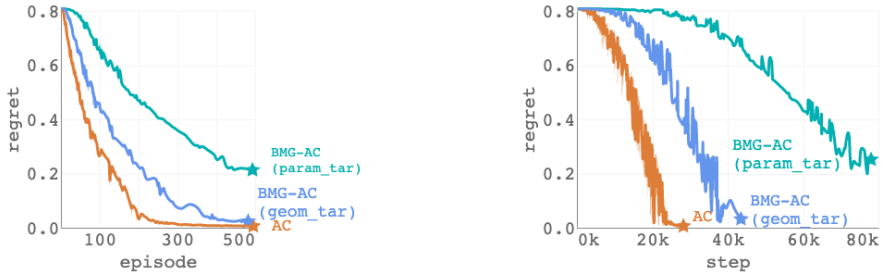
Figure 9: Meta-learner uses Adam. Policy optimization with adaptive optimistic policy gradients. x-axis - (a) no of episodes, (b) no of steps. y-axis - regret $V_\rho(\pi^*) - V_\rho(\pi_t)$. Learning curves denote: the baseline - standard AC algorithm, *optimistic policy gradient learning algorithms* - with parametric target policies , functional non-parametric target policies, trained with meta-gradients from *target predictions*. Shades (wherever noticeable) denote standard error over different runs.



(a) target pred, Adam, total regret

(b) target pred, Adam, final regret

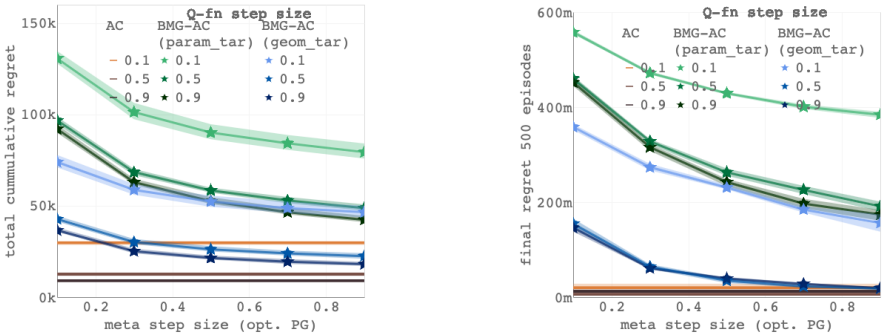
Figure 10: Meta-learner uses Adam. Hyper-parameter sensitivity curves for the meta-learning rate ζ - x-axis. y-axis - total cummulative regret (a): $\sum_{i \leq t} V_\rho(\pi^*) - V_\rho(\pi_i)$, final regret (b): $V_\rho(\pi^*) - V_\rho(\pi_T)$. Learning curves show *optimistic policy gradient learning algorithms* - with parametric target policies , functional non-parametric target policies, trained with meta-gradients from *target predictions*. Different tones show the evolution of the meta-hyperparameter ζ to those used in the inner learned optimization algorithm, i.e. Q-fn step size. Different straight lines denote the baseline—standard AC. Shades denote standard error over different runs.



(a) target pred, SGD, perf/episode

(b) target pred, SGD, perf/step

Figure 11: Meta-learner uses SGD. Policy optimization with optimistic policy gradients. x-axis - (a) no of episodes, (b) no of steps. y-axis - regret $V_\rho(\pi^*) - V_\rho(\pi_t)$. Learning curves denote: the baseline - standard AC algorithm, *optimistic policy gradient learning algorithms* - with parametric target policies, functional non-parametric target policies, trained with meta-gradients from *target predictions*. Shades (wherever noticeable) denote standard error over different runs.



(a) target pred, SGD, total regret

(b) target pred, SGD, final regret

Figure 12: Meta-learner uses SGD. Hyper-parameter sensitivity curves for the meta-learning rate ζ - x-axis; y-axis - total cummulative regret (a): $\sum_{i \leq t} V_\rho(\pi^*) - V_\rho(\pi_i)$, final regret (b): $V_\rho(\pi^*) - V_\rho(\pi_T)$. Learning curves show *optimistic policy gradient learning algorithms* - with parametric target policies , functional non-parametric target policies, trained with meta-gradients from *target predictions*. Different tones show the evolution of the meta-hyperparameter ζ to those used in the inner learned optimization algorithm, i.e. Q-fn step size. Different straight lines denote the baseline—standard AC. Shades denote standard error over different runs.

References

Y. Abbasi-Yadkori, P. Bartlett, K. Bhatia, N. Lazic, C. Szepesvari, and G. Weisz. POLITEX: Regret bounds for policy iteration using expert prediction. In K. Chaudhuri and R. Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 3692–3702. PMLR, 09–15 Jun 2019. URL `https://proceedings.mlr.press/v97/lazic19a.html`.

A. Abdolmaleki, J. T. Springenberg, Y. Tassa, R. Munos, N. Heess, and M. A. Riedmiller. Maximum a posteriori policy optimisation. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018. URL `https://openreview.net/forum?id=S1ANxQW0b`.

A. Agarwal, S. M. Kakade, J. D. Lee, and G. Mahajan. Optimality and approximation with policy gradient methods in markov decision processes. *CoRR*, abs/1908.00261, 2019. URL `http://arxiv.org/abs/1908.00261`.

C. Alfano and P. Rebeschini. Linear convergence for natural policy gradient with log-linear policy parametrization, 2023.

I. Babuschkin, K. Baumli, A. Bell, S. Bhupatiraju, J. Bruce, P. Buchlovsky, D. Budden, T. Cai, A. Clark, I. Danihelka, A. Dedieu, C. Fantacci, J. Godwin, C. Jones, R. Hemsley, T. Hennigan, M. Hessel, S. Hou, S. Kapturowski, T. Keck, I. Kemaev, M. King, M. Kunesch, L. Martens, H. Merzic, V. Mikulik, T. Norman, G. Papamakarios, J. Quan, R. Ring, F. Ruiz, A. Sanchez, R. Schneider, E. Sezener, S. Spencer, S. Srinivasan, W. Stokowiec, L. Wang, G. Zhou, and F. Viola. The DeepMind JAX Ecosystem, 2020. URL `http://github.com/deepmind`.

A. Beck and M. Teboulle. Mirror descent and nonlinear projected subgradient methods for convex optimization. *Oper. Res. Lett.*, 31(3):167–175, 2003. doi: 10.1016/S0167-6377(02)00231-6. URL `https://doi.org/10.1016/S0167-6377(02)00231-6`.

D. P. Bertsekas. Approximate policy iteration: a survey and some new methods. *Journal of Control Theory and Applications*, 9(3):310–335, 2011. doi: 10.1007/s11768-011-1005-3. URL `https://doi.org/10.1007/s11768-011-1005-3`.

D. P. Bertsekas. Weighted sup-norm contractions in dynamic programming: A review and some new applications. 2012.

D. P. Bertsekas and J. N. Tsitsiklis. *Neuro-dynamic programming*. Number 3 in the Optimization and Neural Computation Series. Athena Scientific, Belmont, Mass, 1996.

J. Bhandari and D. Russo. Global optimality guarantees for policy gradient methods. *CoRR*, abs/1906.01786, 2019. URL `http://arxiv.org/abs/1906.01786`.

J. Bhandari and D. Russo. On the linear convergence of policy gradient methods for finite mdps. In A. Banerjee and K. Fukumizu, editors, *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*, volume 130 of *Proceedings of Machine Learning Research*, pages 2386–2394. PMLR, 13–15 Apr 2021. URL `https://proceedings.mlr.press/v130/bhandari21a.html`.

J. Bradbury, R. Frostig, P. Hawkins, M. J. Johnson, C. Leary, D. Maclaurin, G. Necula, A. Paszke, J. VanderPlas, S. Wanderman-Milne, et al. Jax: composable transformations of python+ numpy programs. 2018.

C. Browne, E. J. Powley, D. Whitehouse, S. M. Lucas, P. I. Cowling, P. Rohlfshagen, S. Tavener, D. P. Liebana, S. Samothrakis, and S. Colton. A survey of monte carlo tree search methods. *IEEE Trans. Comput. Intell. AI Games*, 4(1):1–43, 2012. doi: 10.1109/TCIAIG.2012.2186810. URL `https://doi.org/10.1109/TCIAIG.2012.2186810`.

S. Bubeck. Convex optimization: Algorithms and complexity. *Found. Trends Mach. Learn.*, 8(3-4):231–357, 2015. doi: 10.1561/2200000050. URL `https://doi.org/10.1561/2200000050`.

S. Cen, C. Cheng, Y. Chen, Y. Wei, and Y. Chi. Fast global convergence of natural policy gradient methods with entropy regularization. *Oper. Res.*, 70(4):2563–2578, 2022. doi: 10.1287/opre.2021.2151. URL `https://doi.org/10.1287/opre.2021.2151`.

Y. Chandak, G. Theocharous, S. Shankar, M. White, S. Mahadevan, and P. S. Thomas. Optimizing for the future in non-stationary mdps. *CoRR*, abs/2005.08158, 2020. URL `https://arxiv.org/abs/2005.08158`.

Y. Chebotar, A. Molchanov, S. Bechtle, L. Righetti, F. Meier, and G. S. Sukhatme. Meta-learning via learned loss. *CoRR*, abs/1906.05374, 2019. URL `http://arxiv.org/abs/1906.05374`.

Z. Chen and S. Theja Maguluri. Sample complexity of policy-based methods under off-policy sampling and linear function approximation. In G. Camps-Valls, F. J. R. Ruiz, and I. Valera, editors, *Proceedings of The 25th International Conference on Artificial Intelligence and Statistics*, volume 151 of *Proceedings of Machine Learning Research*, pages 11195–11214. PMLR, 28–30 Mar 2022. URL https://proceedings.mlr.press/v151/chen22i.html.

C. Cheng, X. Yan, N. D. Ratliff, and B. Boots. Predictor-corrector policy optimization. *CoRR*, abs/1810.06509, 2018. URL http://arxiv.org/abs/1810.06509.

C.-K. Chiang, T. Yang, C.-J. Lee, M. Mahdavi, C.-J. Lu, R. Jin, and S. Zhu. Online optimization with gradual variations. In S. Mannor, N. Srebro, and R. C. Williamson, editors, *Proceedings of the 25th Annual Conference on Learning Theory*, volume 23 of *Proceedings of Machine Learning Research*, pages 6.1–6.20, Edinburgh, Scotland, 25–27 Jun 2012. PMLR. URL https://proceedings.mlr.press/v23/chiang12.html.

Y. Duan, J. Schulman, X. Chen, P. L. Bartlett, I. Sutskever, and P. Abbeel. Rl^2: Fast reinforcement learning via slow reinforcement learning. *CoRR*, abs/1611.02779, 2016. URL http://arxiv.org/abs/1611.02779.

Y. Efroni, G. Dalal, B. Scherrer, and S. Mannor. Beyond the one step greedy approach in reinforcement learning, 2018a. URL https://arxiv.org/abs/1802.03654.

Y. Efroni, G. Dalal, B. Scherrer, and S. Mannor. Multiple-step greedy policies in online and approximate reinforcement learning. *CoRR*, abs/1805.07956, 2018b. URL http://arxiv.org/abs/1805.07956.

L. Espeholt, H. Soyer, R. Munos, K. Simonyan, V. Mnih, T. Ward, Y. Doron, V. Firoiu, T. Harley, I. Dunning, S. Legg, and K. Kavukcuoglu. IMPALA: scalable distributed deep-rl with importance weighted actor-learner architectures. *CoRR*, abs/1802.01561, 2018. URL http://arxiv.org/abs/1802.01561.

C. Finn, P. Abbeel, and S. Levine. Model-agnostic meta-learning for fast adaptation of deep networks. *CoRR*, abs/1703.03400, 2017. URL http://arxiv.org/abs/1703.03400.

S. Flennerhag, Y. Schroecker, T. Zahavy, H. van Hasselt, D. Silver, and S. Singh. Bootstrapped meta-learning. *CoRR*, abs/2109.04504, 2021. URL https://arxiv.org/abs/2109.04504.

M. Geist, B. Scherrer, and O. Pietquin. A theory of regularized markov decision processes. *CoRR*, abs/1901.11275, 2019. URL http://arxiv.org/abs/1901.11275.

J.-B. Grill, F. Altché, Y. Tang, T. Hubert, M. Valko, I. Antonoglou, and R. Munos. Monte-carlo tree search as regularized policy optimization, 2020. URL https://arxiv.org/abs/2007.12509.

A. Gupta, R. Mendonca, Y. Liu, P. Abbeel, and S. Levine. Meta-reinforcement learning of structured exploration strategies. *CoRR*, abs/1802.07245, 2018. URL http://arxiv.org/abs/1802.07245.

B. Hao, N. Lazic, Y. Abbasi-Yadkori, P. Joulani, and C. Szepesvári. Provably efficient adaptive approximate policy iteration. *CoRR*, abs/2002.03069, 2020. URL https://arxiv.org/abs/2002.03069.

B. Hao, N. Lazic, Y. Abbasi-Yadkori, P. Joulani, and C. Szepesvári. Adaptive approximate policy iteration. In *International Conference on Artificial Intelligence and Statistics*, pages 523–531. PMLR, 2021.

E. Hazan. *Introduction to Online Convex Optimization*. Foundations and Trends in Optimization. Now, Boston, 2017. ISBN 978-1-68083-170-2. doi: 10.1561/2400000013.

T. Hennigan, T. Cai, T. Norman, and I. Babuschkin. Haiku: Sonnet for JAX, 2020. URL http://github.com/deepmind/dm-haiku.

M. Hessel, I. Danihelka, F. Viola, A. Guez, S. Schmitt, L. Sifre, T. Weber, D. Silver, and H. van Hasselt. Muesli: Combining improvements in policy optimization. *CoRR*, abs/2104.06159, 2021. URL https://arxiv.org/abs/2104.06159.

R. Houthooft, R. Y. Chen, P. Isola, B. C. Stadie, F. Wolski, J. Ho, and P. Abbeel. Evolved policy gradients. *CoRR*, abs/1802.04821, 2018. URL http://arxiv.org/abs/1802.04821.

P. Joulani, A. Raj, A. Gyorgy, and C. Szepesvari. A simpler approach to accelerated optimization: iterative averaging meets optimism. In H. D. III and A. Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 4984–4993. PMLR, 13–18 Jul 2020. URL https://proceedings.mlr.press/v119/joulani20a.html.

A. Juditsky, A. S. Nemirovskii, and C. Tauvel. Solving variational inequalities with stochastic mirror-prox algorithm, 2011.

S. Kakade and J. Langford. Approximately optimal approximate reinforcement learning. In *Proceedings of the Nineteenth International Conference on Machine Learning*, ICML '02, page 267–274, San Francisco, CA, USA, 2002. Morgan Kaufmann Publishers Inc. ISBN 1558608737.

S. M. Kakade. A natural policy gradient. In T. Dietterich, S. Becker, and Z. Ghahramani, editors, *Advances in Neural Information Processing Systems*, volume 14. MIT Press, 2001. URL https://proceedings.neurips.cc/paper/2001/file/4b86abe48d358ecf194c56c69108433e-Paper.pdf.

S. Khodadadian, P. R. Jhunjhunwala, S. M. Varma, and S. T. Maguluri. On the linear convergence of natural policy gradient algorithm. *CoRR*, abs/2105.01424, 2021. URL https://arxiv.org/abs/2105.01424.

D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In Y. Bengio and Y. LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL http://arxiv.org/abs/1412.6980.

L. Kirsch, S. van Steenkiste, and J. Schmidhuber. Improving generalization in meta reinforcement learning using learned objectives. *CoRR*, abs/1910.04098, 2019. URL http://arxiv.org/abs/1910.04098.

V. R. Konda and V. S. Borkar. Actor-critic–type learning algorithms for markov decision processes. *SIAM Journal on Control and Optimization*, 38(1):94–123, 1999. doi: 10.1137/S036301299731669X. URL https://doi.org/10.1137/S036301299731669X.

G. M. Korpelevich. The extragradient method for finding saddle points and other problems. 1976.

G. Lan. Policy mirror descent for reinforcement learning: Linear convergence, new sampling complexity, and generalized problem classes, 2022.

N. Lazic, D. Yin, Y. Abbasi-Yadkori, and C. Szepesvari. Improved regret bound and experience replay in regularized policy iteration. In M. Meila and T. Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 6032–6042. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/lazic21a.html.

H. Li, N. Clavette, and H. He. An analytical update rule for general policy optimization. *CoRR*, abs/2112.02045, 2021. URL https://arxiv.org/abs/2112.02045.

J. Luketina, S. Flennerhag, Y. Schroecker, D. Abel, T. Zahavy, and S. Singh. Meta-gradients in non-stationary environments, 2022.

J. Martens. New perspectives on the natural gradient method. *CoRR*, abs/1412.1193, 2014. URL http://arxiv.org/abs/1412.1193.

J. Mei, C. Xiao, B. Dai, L. Li, C. Szepesvári, and D. Schuurmans. Escaping the gravitational pull of softmax. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020a. URL https://proceedings.neurips.cc/paper/2020/hash/f1cf2a082126bf02de0b307778ce73a7-Abstract.html.

J. Mei, C. Xiao, C. Szepesvári, and D. Schuurmans. On the global convergence rates of softmax policy gradient methods. *CoRR*, abs/2005.06392, 2020b. URL https://arxiv.org/abs/2005.06392.

J. Mei, B. Dai, C. Xiao, C. Szepesvári, and D. Schuurmans. Understanding the effect of stochasticity in policy optimization. *CoRR*, abs/2110.15572, 2021a. URL https://arxiv.org/abs/2110.15572.

J. Mei, Y. Gao, B. Dai, C. Szepesvári, and D. Schuurmans. Leveraging non-uniformity in first-order non-convex optimization. *CoRR*, abs/2105.06072, 2021b. URL https://arxiv.org/abs/2105.06072.

J. Mei, W. Chung, V. Thomas, B. Dai, C. Szepesvari, and D. Schuurmans. The role of baselines in policy gradient optimization, 2023.

M. Mohri and S. Yang. Accelerating optimization via adaptive prediction, 2015.

M. Mohri and S. Yang. Accelerating online convex optimization via adaptive prediction. In A. Gretton and C. C. Robert, editors, *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics, AISTATS 2016, Cadiz, Spain, May 9-11, 2016*, volume 51 of *JMLR Workshop and Conference Proceedings*, pages 848–856. JMLR.org, 2016. URL http://proceedings.mlr.press/v51/mohri16.html.

T. Moskovitz, B. O'Donoghue, V. Veeriah, S. Flennerhag, S. Singh, and T. Zahavy. Reload: Reinforcement learning with optimistic ascent-descent for last-iterate convergence in constrained mdps, 2023.

A. Nemirovski. Prox-method with rate of convergence o(1/t) for variational inequalities with lipschitz continuous monotone operators and smooth convex-concave saddle point problems. *SIAM J. Optim.*, 15(1):229–251, 2004. doi: 10.1137/S1052623403425629. URL https://doi.org/10.1137/S1052623403425629.

Y. Nesterov. A method for solving the convex programming problem with convergence rate $o(1/k^2)$. *Proceedings of the USSR Academy of Sciences*, $269 : 543 - -547, 1983$.

M. L. Puterman. *Markov Decision Processes*. Wiley, 1994. ISBN 978-0471727828. URL http://books.google.com/books/about/Markov_decision_processes.html?id=Y-gmAQAAIAAJ.

A. Rakhlin and K. Sridharan. Online learning with predictable sequences. In S. Shalev-Shwartz and I. Steinwart, editors, *COLT 2013 - The 26th Annual Conference on Learning Theory, June 12-14, 2013, Princeton University, NJ, USA*, volume 30 of *JMLR Workshop and Conference Proceedings*, pages 993–1019. JMLR.org, 2013a. URL http://proceedings.mlr.press/v30/Rakhlin13.html.

A. Rakhlin and K. Sridharan. Optimization, learning, and games with predictable sequences. *CoRR*, abs/1311.1869, 2013b. URL http://arxiv.org/abs/1311.1869.

A. Rakhlin and K. Sridharan. Online learning with predictable sequences, 2014.

G. Raskutti and S. Mukherjee. The information geometry of mirror descent, 2014.

B. Scherrer. Improved and generalized upper bounds on the complexity of policy iteration, 2016.

B. Scherrer and C. Thiery. Performance bound for Approximate Optimistic Policy Iteration. Technical report, 2010. URL https://inria.hal.science/inria-00480952.

B. Scherrer, M. Ghavamzadeh, V. Gabillon, B. Lesner, and M. Geist. Approximate modified policy iteration and its application to the game of tetris. *Journal of Machine Learning Research*, 16(49):1629–1676, 2015. URL http://jmlr.org/papers/v16/scherrer15a.html.

J. Schmidhuber. Evolutionary principles in self-referential learning. on learning now to learn: The meta-meta-meta...-hook. Diploma thesis, Technische Universitat Munchen, Germany, 14 May 1987. URL http://www.idsia.ch/~juergen/diploma.html.

J. Schrittwieser, I. Antonoglou, T. Hubert, K. Simonyan, L. Sifre, S. Schmitt, A. Guez, E. Lockhart, D. Hassabis, T. Graepel, T. P. Lillicrap, and D. Silver. Mastering atari, go, chess and shogi by planning with a learned model. *CoRR*, abs/1911.08265, 2019. URL http://arxiv.org/abs/1911.08265.

J. Schulman, S. Levine, P. Moritz, M. I. Jordan, and P. Abbeel. Trust region policy optimization. *CoRR*, abs/1502.05477, 2015. URL http://arxiv.org/abs/1502.05477.

J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. Proximal policy optimization algorithms. *CoRR*, abs/1707.06347, 2017. URL http://arxiv.org/abs/1707.06347.

L. Shani, Y. Efroni, and S. Mannor. Adaptive trust region policy optimization: Global convergence and faster rates for regularized mdps, 2019. URL https://arxiv.org/abs/1909.02769.

D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. van den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel, and D. Hassabis. Mastering the game of go with deep neural networks and tree search. *Nature*, 529(7587):484–489, 2016a. doi: 10.1038/nature16961. URL https://doi.org/10.1038/nature16961.

D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. van den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. P. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel, and D. Hassabis. Mastering the game of go with deep neural networks and tree search. *Nat.*, 529(7587):484–489, 2016b. doi: 10.1038/nature16961. URL https://doi.org/10.1038/nature16961.

D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton, Y. Chen, T. Lillicrap, F. Hui, L. Sifre, G. van den Driessche, T. Graepel, and D. Hassabis. Mastering the game of go without human knowledge. *Nature*, 550(7676):354–359, 2017. doi: 10.1038/nature24270. URL https://doi.org/10.1038/nature24270.

F. Sung, L. Zhang, T. Xiang, T. M. Hospedales, and Y. Yang. Learning to learn: Meta-critic networks for sample efficient learning. *CoRR*, abs/1706.09529, 2017. URL http://arxiv.org/abs/1706.09529.

R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction*. A Bradford Book, Cambridge, MA, USA, 2018. ISBN 0262039249.

R. S. Sutton, D. McAllester, S. Singh, and Y. Mansour. Policy gradient methods for reinforcement learning with function approximation. In *Proceedings of the 12th International Conference on Neural Information Processing Systems*, NIPS'99, page 1057–1063, Cambridge, MA, USA, 1999. MIT Press.

M. Tomar, L. Shani, Y. Efroni, and M. Ghavamzadeh. Mirror descent policy optimization. *CoRR*, abs/2005.09814, 2020. URL `https://arxiv.org/abs/2005.09814`.

J. N. Tsitsiklis. On the convergence of optimistic policy iteration. *J. Mach. Learn. Res.*, 3:59–72, 2002. URL `http://jmlr.org/papers/v3/tsitsiklis02a.html`.

V. Veeriah, M. Hessel, Z. Xu, R. L. Lewis, J. Rajendran, J. Oh, H. van Hasselt, D. Silver, and S. Singh. Discovery of useful questions as auxiliary tasks. *CoRR*, abs/1909.04607, 2019. URL `http://arxiv.org/abs/1909.04607`.

N. Vieillard, T. Kozuno, B. Scherrer, O. Pietquin, R. Munos, and M. Geist. Leverage the average: an analysis of regularization in RL. *CoRR*, abs/2003.14089, 2020a. URL `https://arxiv.org/abs/2003.14089`.

N. Vieillard, B. Scherrer, O. Pietquin, and M. Geist. Momentum in reinforcement learning. In S. Chiappa and R. Calandra, editors, *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pages 2529–2538. PMLR, 26–28 Aug 2020b. URL `https://proceedings.mlr.press/v108/vieillard20a.html`.

P. Wagner. Optimistic policy iteration and natural actor-critic: A unifying view and a non-optimality result. In C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc., 2013. URL `https://proceedings.neurips.cc/paper_files/paper/2013/file/2dace78f80bc92e6d7493423d729448e-Paper.pdf`.

P. Wagner. Policy oscillation is overshooting. *Neural Networks*, 52:43–61, 2014. doi: 10.1016/j.neunet.2014.01.002. URL `https://doi.org/10.1016/j.neunet.2014.01.002`.

J. Wang and J. D. Abernethy. Acceleration through optimistic no-regret dynamics. *CoRR*, abs/1807.10455, 2018. URL `http://arxiv.org/abs/1807.10455`.

J. Wang, J. D. Abernethy, and K. Y. Levy. No-regret dynamics in the fenchel game: A unified framework for algorithmic convex optimization. *CoRR*, abs/2111.11309, 2021. URL `https://arxiv.org/abs/2111.11309`.

Y. Wang, Q. Ye, and T. Liu. Beyond exponentially discounted sum: Automatic learning of return function. *CoRR*, abs/1905.11591, 2019. URL `http://arxiv.org/abs/1905.11591`.

R. J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Mach. Learn.*, 8:229–256, 1992. doi: 10.1007/BF00992696. URL `https://doi.org/10.1007/BF00992696`.

L. Xiao. On the convergence rates of policy gradient methods, 2022.

Z. Xu, H. van Hasselt, and D. Silver. Meta-gradient reinforcement learning. In S. Bengio, H. M. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 2402–2413, 2018. URL `https://proceedings.neurips.cc/paper/2018/hash/2715518c875999308842e3455eda2fe3-Abstract.html`.

Z. Xu, H. van Hasselt, M. Hessel, J. Oh, S. Singh, and D. Silver. Meta-gradient reinforcement learning with an objective discovered online. *CoRR*, abs/2007.08433, 2020. URL `https://arxiv.org/abs/2007.08433`.

Y. Ye. The simplex and policy-iteration methods are strongly polynomial for the markov decision problem with a fixed discount rate. *Mathematics of Operations Research*, 36(4):593–603, 2011. ISSN 0364765X, 15265471. URL `http://www.jstor.org/stable/41412326`.

R. Yuan, S. S. Du, R. M. Gower, A. Lazaric, and L. Xiao. Linear convergence of natural policy gradient methods with log-linear policies. In *The Eleventh International Conference on Learning Representations*, 2023. URL `https://openreview.net/forum?id=-z9hdsyUwVQ`.

T. Zahavy, Z. Xu, V. Veeriah, M. Hessel, J. Oh, H. van Hasselt, D. Silver, and S. Singh. A self-tuning actor-critic algorithm. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL `https://proceedings.neurips.cc/paper/2020/hash/f02208a057804ee16ac72ff4d3cec53b-Abstract.html`.

Z. Zheng, J. Oh, and S. Singh. On learning intrinsic rewards for policy gradient methods. *CoRR*, abs/1804.06459, 2018. URL `http://arxiv.org/abs/1804.06459`.

M. Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *Proceedings of the Twentieth International Conference on International Conference on Machine Learning*, ICML'03, page 928–935. AAAI Press, 2003. ISBN 1577351894.