Functional Acceleration for Policy Mirror Descent

Veronica Chelu[❀, ✿] and Doina Precup[❀, ✿, ❁, *]

[❀]McGill University, [✿]Mila Quebec AI Institute, [❁]Google DeepMind, [*]CIFAR AI Chair

Abstract

We apply *functional acceleration* to the Policy Mirror Descent (PMD) general family of algorithms, which cover a wide range of novel and fundamental methods in Reinforcement Learning (RL). Leveraging duality, we propose a momentum-based PMD update. By taking the functional route, our approach is independent of the policy parametrization and applicable to large-scale optimization, covering previous applications of momentum at the level of policy parameters as a special case. We theoretically analyze several properties of this approach and complement with a numerical ablation study, which serves to illustrate the policy optimization dynamics on the value polytope, relative to different algorithmic design choices in this space. We further characterize numerically several features of the problem setting relevant for functional acceleration, and lastly, we investigate the impact of approximation on their learning mechanics [1].

1 Introduction

The RL framework (Sutton and Barto, 2018) refers to the problem of solving sequential decision making tasks under uncertainty, together with a class of solution methods tailored for it. The RL problem has found applications in games (Tesauro, 1994; Mnih et al., 2013; Silver et al., 2014; Mnih et al., 2016; Silver et al., 2017; Hessel et al., 2017; Bellemare et al., 2017; Schrittwieser et al., 2019; Zahavy et al., 2023), robotic manipulation (Schulman et al., 2015, 2017; Haarnoja et al., 2018), medicine (Jumper et al., 2021; Schaefer et al., 2004; Nie et al., 2020) and is formally described by means of discounted Markov Decision Processes (MDPs) (Puterman, 1994). On the solution side, increased interest has been devoted to the study of policy-gradient (PG) approaches based on optimizing a parameterised policy with respect to an objective (Williams, 1992; Konda and Borkar, 1999; Sutton et al., 1999; Agarwal et al., 2019; Bhandari and Russo, 2019; Kakade, 2001; Bhandari and Russo, 2021; Mei et al., 2020b,a).

Policy Mirror Descent (PMD) (Agarwal et al., 2019; Bhandari and Russo, 2021; Xiao, 2022; Johnson et al., 2023; Vaswani et al., 2021) is a general family of algorithms, specified by the choice of mirror map covering a wide range of novel and fundamental methods in RL. PMD is a proximal algorithm (Parikh et al., 2014) and an instance of Mirror Descent (MD) (Beck and Teboulle, 2003) on the policy simplex (Bhandari and Russo, 2021), which applies a proximal regularization to the improvement step of Policy Iteration (PI), and converges to it as regularization decreases. In the γ-discounted setting, with an adaptive step-size, it converges linearly at the optimal γ-rate, independent of the dimension of the state space or problem instance (Johnson et al., 2023), recovering classical approaches, like PI and VI, as special cases. PMD has been extended to linear approximation by Yuan et al. (2023) and to general function approximation by Alfano et al. (2024). The latter uses the L_2-norm to measure the function approximation error and applies the PMD update in the dual form, as (generalized) Projected Gradient Descent (PGD), i.e. a gradient update in the dual space followed by a projection (Bubeck, 2015), rather than in proximal form (Beck and Teboulle, 2003), as extended by Tomar et al. (2020), and later analyzed by Vaswani et al. (2021), who treat the PMD surrogate objective as a nonlinear

[1]Code is available at https://github.com/veronicachelu/functional-acceleration-for-pmd

Workshop on Aligning Reinforcement Learning Experimentalists and Theorists (ARLET 2024).

optimization problem, that of approximately minimizing at each iteration, a composite proximal objective, denoted $\ell(\pi^\theta)$ with respect to the policy parameter θ. Vaswani et al. (2023) further relies on a dual policy norm, induced by the chosen mirror map of an approximate PMD update, to measure the critic's evaluation error in a decision-aware actor-critic algorithm. Similarly, we too leverage critic differences in dual space to gain momentum and accelerate the optimization.

Motivation The running time of PMD algorithms scales with the number of iterations. In addition, with a parametrized policy class, each iteration of an approximate PMD method may become sample-inefficient, requiring multiple "inner-loop" updates to the policy parameter (e.g., Vaswani et al. (2021)). Actor-critic (AC) methods (Sutton et al., 1999; Konda and Borkar, 1999) additionally require the computation of an inexact critic corresponding to the action-value function, which may further increase the sample complexity per iteration. It is therefore desirable to design algorithms which converge in a smaller number of iterations, resulting in significant empirical speedups, as has been previously argued by Johnson et al. (2023); Xiao (2022); Goyal and Grand-Clement (2021); Russo (2022).

In this work, we leverage duality and acceleration to build a novel surrogate objective for momentum-based PMD, leading to faster learning in terms of less iterations necessary to converge. The novelty of our approach is the application of acceleration mechanics to the direct or functional policy representation π—hence named *functional acceleration*, as opposed to classic acceleration applied to the policy parameter θ (e.g., Mnih et al. (2016); Hessel et al. (2017); Schulman et al. (2017) use Adam (Kingma and Ba, 2015) or RMSProp (Hinton et al., 2012)). Specifically, we use momentum in the dual policy space to accelerate on "long ravines" or decelerate at "sharp curvatures" at the functional level of the policy optimization objective. Intuitively, adding momentum to the functional PG (the gradient of the policy performance objective with respect to the direct policy representation π) means applying, to the current directional policy derivative, a weighted version of the previous policy ascent direction, encouraging the method to adaptively accelerate according to the geometry of the optimization problem.

Contributions

- ✿ We illustrate and analyze theoretically the impact of applying functional acceleration on the optimization dynamics of PMD, leading to a practical momentum-based PMD algorithm.
- ❀ We characterize the properties of the problem setting, and those intrinsic to the algorithm, for which applying functional acceleration is conducive to faster learning.
- ❁ We study the influence of an inexact critic on the acceleration mechanism proposed.

Outline This document is organized as follows. After placing our work in existing literature in Sec. 2, and setting up the context in which it operates in Sec. 3, we introduce our main ideas in Sec. 4. We complement with numerical studies in Sec. 5, ending with a short closing in Sec.6.

2 Related Work

Accelerated optimization methods have been at the heart of convex optimization research, e.g., 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, 2013; Joulani et al., 2020b), AO-FTRL (Rakhlin and Sridharan, 2014; Mohri and Yang, 2015), Forward-Backward-Forward (FBF) method (Tseng, 1991).

As far as we know, our idea of applying acceleration to the direct (functional) policy representation π^θ—independent of the policy parametrization θ—is novel. This is important because it means universality of the approach to any kind of parameterization and functional form a practitioner requires. Within the context of RL, acceleration has only been applied to value learning (Vieillard et al., 2019; Farahmand and Ghavamzadeh, 2021; Goyal and Grand-Clement, 2021), or in the context of PG methods, classic acceleration is applied to the policy parameter θ—all recent deep RL works (e.g. Mnih et al. (2016); Hessel et al. (2017); Schulman et al. (2017)) use some form of adaptive gradient method, like Adam (Kingma and Ba, 2015) or RMSProp (Hinton et al., 2012). The idea of acceleration generally relies on convexity of the objective relative to the representation of interest. The transformation from parameters θ to functional representation of the policy as probabilities π^θ, can be highly complex, non-linear, and problem-dependent. Proximal algorithms operate on this

functional representation, and rely on relative-convexity and relative-smoothness (Lu et al., 2017)[2] of the objective with respect to π when constructing surrogate models (Bhandari and Russo, 2019, 2021; Agarwal et al., 2019; Vaswani et al., 2021). These properties suggests the functional acceleration mechanism is feasible and promising in our setting, since it is able to successfully accelerate convex optimization (Joulani et al., 2020b).

Limitations & Future Work Our focus is on developing a foundation that motivates further study. A translation to practical large-scale implementations and deep RL remains for further investigation, i.e. with non-standard proximal methods, e.g., TRPO (Schulman et al., 2015), PPO (Schulman et al., 2017), MDPO (Tomar et al., 2020), MPO (Abdolmaleki et al., 2018)). Additional guarantees of accelerated convergence for general policy parametrizations using the dual policy norm, as well as theoretical analysis for the stochastic setting, are also deferred for future work.

3 Background & Preliminaries

RL We consider a standard RL setting described by means of a Markov decision process (MDP) $(\mathcal{S}, \mathcal{A}, r, P, \gamma, \rho)$, with state space $\mathcal{S}$, action space $\mathcal{A}$, discount factor $\gamma \in [0, 1)$, initial state distribution $\rho \in \Delta(S)$ ($\Delta(\mathcal{X})$—the probability simplex over a set $\mathcal{X}$), rewards are sampled from a reward function $R \sim r(S, A), r : \mathcal{S} \times \mathcal{A} \to [0, R_{\max}]$, and next states from a transition probability distribution $S' \sim P(\cdot|S, A) \in \Delta(\mathcal{S})$. The RL problem (Sutton and Barto, 2018) consists in finding a policy $\pi : \mathcal{S} \to \Delta_{\mathcal{A}} \in \Pi = \Delta_{\mathcal{A}}^{|\mathcal{S}|}$, maximizing the performance objective defined as the discounted expected cumulative reward $V_\rho^\pi \doteq \mathbb{E}_{s\sim\rho} V_s^\pi \in \mathbb{R}$, where $V^\pi \in \mathbb{R}^{|\mathcal{S}|}$ and $Q^\pi \in \mathbb{R}^{|\mathcal{S}|\times|\mathcal{A}|}$ are the value and action-value functions of a policy π, such that $V_s^\pi = \mathbb{E}^\pi[\sum_{i=0}^\infty \gamma^i R_{i+1}|S_0 = s]$, $Q_{s,a}^\pi \doteq \mathbb{E}_\pi\left[\sum_{i=0}^\infty \gamma^i R_i|S_0 = s, A_0 = a\right]$ and $V_s^\pi \doteq \mathbb{E}_\pi[Q(s, A)]$. There exists an optimal deterministic policy π^* that simultaneously maximises V^π and Q^π (Bellman, 1957). Let d^π be the discounted visitation distribution $d_s^\pi = (1-\gamma)\sum_{i=0}^\infty \gamma^i \Pr(S_i = s|S_0 \sim \rho, A_j \sim \pi_{s_j}, \forall j \leq i)$.

We use the shorthand notation $\langle\cdot,\cdot\rangle$—the dot product, $\nabla f(x) \doteq \nabla_x f(x)$—gradients and partial derivatives, $\nabla f(x, y) \doteq \nabla_x f(x, y)$, $\pi^t \doteq \pi^{\theta_t}$, $Q^t \doteq Q^{\pi_t}$, $V^t \doteq V^{\pi_t}$, $d_\rho^t \doteq d_\rho^{\pi^t}$, $\pi_s \doteq \pi(\cdot|s)$, $Q_s \doteq Q(s)$, $V_s \doteq V(s)$, $r_s \doteq r(s)$, $\pi_{a|s} \doteq \pi(a|s)$, $Q_{s,a} \doteq Q(s, a)$.

PG Algorithms update the parameters $\theta \in \Theta$ of a parametric policy π^θ using surrogate objectives that are local approximations of the original performance. In the tabular setting, the direct parameterisation associates a parameter to each state-action pair, allowing the shorthand notation $\pi \doteq \pi^\theta$. The gradient of the performance V_ρ^π with respect to the *direct representation* π (Sutton et al., 1999; Agarwal et al., 2019; Bhandari and Russo, 2019)—which we call the *"functional" gradient*, to distinguish it from the gradient $\nabla_\theta V_\rho^{\pi_\theta}$ relative to a policy parameter θ, is $\nabla_{\pi_s} V_\rho^\pi = {}^1\!/_{(1-\gamma)} d_{s,\rho}^\pi Q_{s,a}^\pi \in \mathbb{R}^{|\mathcal{A}|}$. Then, we define $\nabla V_\rho^\pi \in \mathbb{R}^{|\mathcal{S}|\times|\mathcal{A}|}$ as the concatenation of $\nabla_{\pi_s} V_\rho^\pi, \forall s \in \mathcal{S}$ (yielding a PGT for directional derivatives in Lemma 2 in Appendix B.1.1).

Mirror Descent (MD) is a general gradient descent algorithm, applicable to constrained spaces $\mathcal{C}$, which relies on Fenchel conjugate duality to map the iterates of an optimization problem $x^* = \operatorname{argmin}_{x\in\mathcal{X}\cap\mathcal{C}} f(X)$, back and forth between a primal $\mathcal{X}$ and a dual space $\mathcal{X}^*$. The algorithm uses a convex function of the Legendre-type [3], called a mirror map h, to map the MD iterates x to the dual space where the gradient update is performed $\nabla h(y) \doteq \nabla h(x) - \eta\nabla f(x)$, with η a step size. A new iterate satisfying the primal constraints $\mathcal{C}$ is obtained using a Bregman projection $x' \doteq \operatorname{proj}_{\mathcal{C}}^h(y) = \operatorname{argmin}_{x\in\mathcal{C}} D_h(x, \nabla h^*(\nabla h(y)))$ of the updated dual iterate $\nabla h(y)$ mapped back in the primal space using the conjugate function of h, $h^*(x^*) = \sup_{x\in\mathcal{X}}\langle x, x^*\rangle - h(x)$. This projection relies on a Bregman divergence $D_h(x, y) \doteq h(x) - h(y) - \langle\nabla h(y), x - y\rangle$ (Amari, 1998; Bubeck, 2015; Banerjee et al., 2005). Furthermore, cf. Amari (2016), the divergences derived from the two convex functions are substantially the same, except for the order $D_h(x, \nabla h^*(\nabla h(y))) = D_{h^*}(\nabla h(y), \nabla h^*(x))$. The proximal formulation of mirror descent merges the update and projection steps of MD to $x' \doteq \operatorname{argmin}_{\bar{x}\in\mathcal{X}\cap\mathcal{C}} \eta\langle\nabla f(x), \bar{x}\rangle + D_h(\bar{x}, x)$ (see Lemma 3 in Appendix B.1).

[2] relative to the mirror map h.

[3] We require h to be strictly convex and essentially smooth (differentiable and $\|\nabla h(x^t)\| \to \infty$ for any sequence x^t converging to a point on the boundary of $\operatorname{dom} h$) on the relative interior (rint) of $\operatorname{dom} h$.

PMD is an instance of MD (Beck and Teboulle, 2003), applying GD in a non-Euclidean geometry, using the proximal perspective of MD, $\pi^{t+1} \doteq \operatorname{argmin}_{\pi \in \Pi} -\langle \nabla V_\rho^t, \pi \rangle + {}^1\!/\!_{\eta^t} D_h(\pi, \pi^t)$ for some sequence of step-sizes $\eta^t > 0$ and initial policy π^0. The visitation-distribution d_ρ^t in the gradient of the surrogate objective can lead to vanishing gradients in infrequently visited states under π^t (Mei et al., 2020a; Bhandari and Russo, 2021; Johnson et al., 2023), so PMD iteratively applies a variant that separates the objective per state

$$\pi_s^{t+1} \doteq \operatorname{argmin}_{\pi_s \in \Delta(\mathcal{A})} -\langle \widehat{Q}_s^t, \pi_s \rangle + {}^1\!/\!_{\eta^t} D_h(\pi_s, \pi_s^t)$$

where $\widehat{Q}_s^t$ corresponds to Q_s^t (a preconditioned gradient cf. Kakade (2001)) or some approximation thereof, for the exact and inexact versions, respectively. Using the negative Boltzmann-Schannon entropy (Shannon, 1948) as mirror map yields the Natural Policy Gradient (NPG) (Kakade, 2001). With a null Bregman divergence, it recovers PI (Johnson et al., 2023).

Approximate PMD The standard PMD algorithm is adapted by Tomar et al. (2020) and Vaswani et al. (2021) to general policy parametrizations π^θ, by updating the parameters θ using the PMD surrogate objective, which can be expressed as a composite objective $\theta_{t+1} = \operatorname{argmin}_{\theta \in \Theta} \mathbb{E}_{s \sim d_\rho^t}[-\mathbb{E}_{a \sim \pi_s^\theta}[\widehat{Q}_{s,a}^t] + {}^1\!/\!_{\eta^t} D_h(\pi_s^\theta, \pi_s^t)]$. Alfano et al. (2024) introduces the concept of Bregman policy class $\{\pi^\theta : \pi_s^\theta = \operatorname{proj}^h_{\Delta(\mathcal{A})}(\nabla h^*(f_s^\theta)), s \in \mathcal{S}\}$, and uses a parametrized function f^θ to approximate the dual update of MD $f_s^{t+1} \doteq \nabla h(\pi_s^t) - \eta^t \widehat{Q}_s^t$. To satisfy the simplex constraint, a Bregman projection is used on the dual approximation mapped back to the policy space $\pi_s^\theta = \operatorname{proj}^h_{\Delta(\mathcal{A})}(\nabla h^*(f_s^{t+1}))$, equivalent to $\theta_{t+1} = \operatorname{argmin}_{\theta \in \Theta} D_h(\pi_s^\theta, \nabla h^*(f_s^{t+1}))$. Using the negative Boltzmann-Shannon entropy, yields the softmax policy class $\pi_{s,a}^\theta \doteq {}^{\exp f_{s,a}^\theta}\!/\!_{\| \exp f_s^\theta \|_1}, \forall s, a \in \mathcal{S} \times \mathcal{A}$.

4 Functional Acceleration for PMD

In this work, we primarily focus on a momentum-based PMD update. To build some intuition around the proposed update, consider first an idealized update, called `PMD(+lookahead)`, anticipating one iteration ahead on the optimization path using the lookhead return $\tilde{Q}_s^t, \forall s \in \mathcal{S}$

$$\tilde{\pi}_s^t = \operatorname{greedy}(\widehat{Q}_s^t) \qquad \tilde{Q}_s^t = \mathbb{E}[r_s + \gamma \langle \widehat{Q}_{s'}^t, \tilde{\pi}_{s'}^t \rangle] \tag{1}$$

where $\tilde{Q}_s^t$ is the expected return of acting greedily with $\tilde{\pi}^t$ for one iteration, and following π^t thereafter. With $\tilde{\eta}^t$ an adaptive step-size, it leads to the PGD update $\pi^{t+1} = \operatorname{argmin}_{\pi_s \in \Delta(\mathcal{A})} D_h(\pi_s, \nabla h^*(\nabla h(\pi_s^t) - \tilde{\eta}^t \tilde{Q}_s^t))$ (cf. Alfano et al. (2024)), and to the proximal update (cf. Johnson et al. (2023))

$$\pi_s^{t+1} = \operatorname{argmin}_{\pi_s \in \Delta(\mathcal{A})} -\langle \tilde{Q}_s^t, \pi_s \rangle + {}^1\!/\!_{\tilde{\eta}^t} D_h(\pi_s, \pi_s^t) \tag{2}$$

Prop. 1 indicates `PMD(+lookahead)` (Eq. 1 & Eq. 2) accelerates convergence by changing the contraction rate via the discount factor γ^2, instead of the traditional γ, corresponding to the one-step lookahead horizon chosen here, $H = 1$ (generalizing to γ^{H+1} for multi-step). Proof in Appendix B.2.

Proposition 1. *(Functional acceleration with exact* `PMD(+lookahead)`*) The policy iterates* π^{t+1} *of* `PMD(+lookahead)` *satisfy* $\|V^* - V^t\|_\infty \leq (\gamma^2)^t (\|V^* - V^0\|_\infty + \sum_{i \leq t} {}^{\epsilon_i}\!/\!_{(\gamma^2)^i})$*, with step-size adaptation,* $\tilde{\eta}^t \geq {}^1\!/\!_{\epsilon_t} D_h(\operatorname{greedy}(\tilde{Q}_s^t), \pi_s^t)$*,* $\forall \epsilon_t$ *arbitrarily small.*

For inexact critics $\widehat{Q}^t$, it is known that the hard greedification operator $\tilde{\pi}_s^t = \operatorname{greedy}(\widehat{Q}_s^t)$ can yield unstable updates. Taking inspiration from the "mirror prox" method of Nemirovski (2004) (aka the "extragradient" (extrapolated gradient) method), we further relax the lookahead by replacing the hard greedification in Eq. 1 with another PMD update

$$\tilde{\pi}_s^t = \operatorname{argmin}_{\pi_s \in \Delta(\mathcal{A})} -\langle \widehat{Q}_s^t, \pi_s \rangle + {}^1\!/\!_{\eta^t} D_h(\pi_s, \pi_s^t) \tag{3}$$

We denote `PMD(+extragradient)` the combination of Eq. 3 & 2. Prop. 2 confirms the acceleration property of `PMD(+lookahead)` is maintained.

Proposition 2. *(Functional acceleration with* `PMD(+extragradient)`*) The policy iterates* π^{t+1} *of* `PMD(+extragradient)` *satisfy* $\|V^* - V^t\|_\infty \leq (\gamma^2)^t (\|V^* - V^0\|_\infty + \sum_{i \leq t} {}^{(\epsilon_i + \gamma \tilde{\epsilon}_t)}\!/\!_{(\gamma^2)^i})$*, with step-size adaptation,* $\eta^t \geq {}^1\!/\!_{\tilde{\epsilon}_t} D_h(\operatorname{greedy}(\widehat{Q}_s^t), \pi_s^t)$*, and* $\tilde{\eta}^t \geq {}^1\!/\!_{\epsilon_t} D_h(\operatorname{greedy}(\widetilde{Q}_s^t), \pi_s^t)$*,* $\forall \epsilon_t, \tilde{\epsilon}_t$ *arbitrarily small.*

This algorithm uses intermediary policies $\tilde{\pi}_s^t$ to look ahead, but the next policy iterate is obtained from π_s^t, so it requires keeping two policies for each iteration. The next proposition shows that the solutions π_s^{t+1} of `PMD(+extragradient)` subsume those of another update, called `PMD(+correction)` (Eq. 3 & Eq. 4), which relaxes this requirement by obtaining the next policy directly from $\tilde{\pi}^t$. It does this by using a lookahead correction $\tilde{\eta}^t\tilde{Q}_s^t - \eta^t\widehat{Q}_s^t$ (rather than using the lookahead $\tilde{\eta}^t\tilde{Q}_s^t$). The PGD update is $\pi^{t+1} = \operatorname{argmin}_{\pi_s\in\Delta(\mathcal{A})} D_h(\pi_s, \nabla h^*(\nabla h(\tilde{\pi}_s^t) - [\tilde{\eta}^t\tilde{Q}_s^t - \eta^t\widehat{Q}_s^t]))$, whereas the proximal perspective is

$$\pi_s^{t+1} = \operatorname{argmin}_{\pi_s\in\Delta(\mathcal{A})} -\langle \tilde{Q}_s^t - {}^{\eta^t}\!/_{\tilde{\eta}^t}\widehat{Q}_s^t, \pi_s\rangle + {}^{1}\!/_{\tilde{\eta}^t} D_h(\pi_s, \tilde{\pi}_s^t) \tag{4}$$

Proposition 3. *(Extrapolation from the future) The solutions of* `PMD(+extragradient)` *subsume those of* `PMD(+correction)`.

The update of `PMD(+correction)` in Eq. 4 is a relaxation of the update of `PMD(+extragradient)` in the sense that we no longer need to rollback to π^t to perform the update, rather, we can use the freshest policy iterate available, which is $\tilde{\pi}^t$, keeping around a single policy at a time. However, this also means potentially missing out some solutions due to the extra intermediary projection via $\tilde{\pi}^t$. This update takes inspiration from the "forward-backward-forward" and "predictor-corrector" methods of of Tseng (1991), and Cheng et al. (2018), respectively.

At this point, we still need two evaluations per iteration, that of the action-value function $\widehat{Q}_s^t \approx Q_s^t$, and the lookahead $\tilde{Q}_s^t$, which is inefficient without model-based access. The following *lazy* counterparts further relax this assumption, using a single evaluation per iteration, at the expense of extra memory, performing *"extrapolation from the past"* (Gidel et al., 2018; Böhm et al., 2020), by delaying the correction and recycling previous Q-functions. In other words, we may apply the correction for timestep $t-1$ with delay, at timestep t, and we may use a single set of Q-function corresponding to the policy sequence $\{\pi^t\}_{t\geq 0}$, leading to an update called `Lazy PMD(+correction)`(Eq. 5 & Eq. 6)

$$\tilde{\pi}_s^t = \operatorname{argmin}_{\pi_s\in\Delta(\mathcal{A})} -\langle {}^{\eta^{t-1}}\!/_{\eta_t}(\widehat{Q}_s^t - \widehat{Q}_s^{t-1}), \pi_s\rangle + {}^{1}\!/_{\eta^{t-1}} D_h(\pi_s, \pi_s^t) \tag{5}$$

$$\pi_s^{t+1} = \operatorname{argmin}_{\pi_s\in\Delta(\mathcal{A})} -\langle \widehat{Q}_s^t, \pi_s\rangle + {}^{1}\!/_{\eta^t} D_h(\pi_s, \tilde{\pi}_s^t) \tag{6}$$

Finally, by relying on a single set of policy iterates, we may merge the two updates in Eq. 5 and 6, into a momentum-based PMD update, denoted `Lazy PMD(+momentum)` (Eq. 7),

$$\pi_s^{t+1} = \operatorname*{argmin}_{\pi_s\in\Delta(\mathcal{A})} -\langle \widehat{Q}_s^t + {}^{\eta^{t-1}}\!/_{\eta^t}(\widehat{Q}_s^t - \widehat{Q}_s^{t-1}), \pi_s\rangle + {}^{1}\!/_{\eta^t} D_h(\pi_s, \pi_s^t) \tag{7}$$

An update akin to Eq. 7 is called "optimistic" mirror descent by Joulani et al. (2020b,a); Rakhlin and Sridharan (2013, 2014) and a "forward-reflected-backward" method by Malitsky and Tam (2020). Prop. 4 shows the iterates of `Lazy PMD(+momentum)` fortunately subsume those of `Lazy PMD(+correction)`.

Proposition 4. *(Extrapolation from the past) The solutions of* `Lazy PMD(+momentum)` *subsume those of* `Lazy PMD(+correction)`.

Alg. 2 in Appendix C summarizes the aforementioned updates.

4.1 Approximate Functional Acceleration for Parametric Policies

We are interested in designing algorithms feasible for large-scale optimization, so we further consider parametrized versions of the functional acceleration algorithms introduced, which we illustrate numerically in Sec. 5.

Q-function Approximation For the *exact setting*, we compute model-based versions of all updates, $\widehat{Q}^t \doteq Q^t$. For the *inexact setting*, we consider approximation errors between $\widehat{Q}^t$ and Q^t (Sec.5.3).

Algorithm 1 `Approximate Lazy PMD(+momentum)`

Initialize policy parameter $\theta_0 \in \Theta$, mirror map h, small constant ϵ_0, learning rate β
for $t = 1, 2 \dots T$ **do**
  Find $\widehat{Q}^t$ approximating Q^t (critic update)
  Compute adaptive step-size $\eta^t = {}^{D_h(\text{greedy}(\widehat{Q}^t), \pi^t)}/_{\gamma^{2(t+1)}\epsilon_0}$
  Find $\pi^{t+1} \doteq \pi^{\theta_{t+1}}$ by solving the surrogate problem (approximately with k GD updates)
    $\min_{\theta \in \Theta} \ell(\theta) \qquad \ell(\theta) \doteq -\mathbb{E}_{s \sim d_\rho^t}[\mathbb{E}_{a \sim \pi^\theta}[\widehat{Q}^t_{s,a} + {}^{\eta_{t-1}}/_{\eta_t}(\widehat{Q}^t_{s,a} - \widehat{Q}^{t-1}_{s,a})] + {}^1/_{\eta^t} D_h(\pi^\theta_s, \pi^t_s)]$
    (init) $\theta^{(0)} \doteq \theta_t \qquad$ *(for* $i \in [0..k-1]$*)* $\theta^{(i+1)} = \theta^{(i)} - \beta \nabla_{\theta^{(i)}} \ell(\theta^{(i)}) \qquad$ *(final)* $\theta_{t+1} \doteq \theta^{(k)}$
end for

Policy Approximation We parametrize the policy iterates using a Bregman policy class $\{\pi^\theta : \pi^\theta_s = \text{proj}^h_{\Delta(\mathcal{A})}(\nabla h^*(f^\theta_s)), s \in \mathcal{S}\}$, a tabular parametrization for the dual policy representation $f^\theta_{s,a} \doteq \theta_{s,a}$, and the negative Boltzmann-Shannon entropy as mirror map h, which leads to the softmax policy class $\pi^\theta \doteq {}^{\exp \theta_{s,\cdot}}/_{\sum_{a \in \mathcal{A}} \exp \theta_{s,a}}$. There are two ways of updating the parameter vector θ (cf. Lemma 3 (Bubeck, 2015)): (i) the PGD perspective of MD (Alfano et al., 2024; Haarnoja et al., 2018; Abdolmaleki et al., 2018) (see Appendix E), or (ii) the proximal perspective (Tomar et al., 2020; Vaswani et al., 2021, 2023). The latter is used and described in Alg. 1. We execute the parameter optimization in Alg. 1 in expectation over the state-action space—in full-batch (computing d^π_ρ exactly and in expectation for all actions) to showcase the higher-level optimization that is the spotlight of this work and remove any other collateral artifacts or confounding effects from exploration of the state space or too early committal to a strategy (Mei et al., 2021). Practical large-scale algorithms apply mini-batches sampled from a reply buffer, with the updates somewhere between full-batch and online. In making this simplification, we inevitably leave complementary investigations on the influence of stochasticity and variance of the policy gradient for future work. The rest of the algorithms use surrogate objectives cf. Sec. 4 (details in Appendix D).

5 Numerical Studies

In this section, we investigate numerically the aforementioned algorithms, focusing on the following questions: (i) *When is acceleration possible?*—Sec. 5.1 investigates for which settings is functional acceleration opportune, and attempts to characterize properties of the problem which make it advantageous. (ii) *What are the policy optimization dynamics of each functional acceleration method?*—Sec. 5.2 illustrates the policy optimization dynamics of the methods introduced on the space of policy value. (iii) *Should we expect acceleration to be effective with an inexact critic?*—Sec. 5.3 investigates the implications of using value approximation.

5.1 When is Acceleration Possible?

Experimental Setting We consider randomly constructed finite MDPs—`Random MDP` problems (Archibald et al., 1995), abstract, yet representative of the kind of MDP encountered in practice, which serve as a test-bench for RL algorithms (Goyal and Grand-Clement, 2021; Scherrer and Geist, 2014; Vieillard et al., 2019). A `Random MDP` generator $\mathcal{M} \doteq (|\mathcal{S}|, |\mathcal{A}|, b, \gamma)$ is parameterized by 4 parameters: number of states $|\mathcal{S}|$, number of actions $|\mathcal{A}|$, branching factor b specifying for each state-action pair the maximum number of possible next states, chosen randomly. We vary b, γ, and $|\mathcal{A}|$ to show how the characteristics of the problem, and the features of the algorithms, impact learning speed with or without functional acceleration. Additional details in Appendix G.2.

Metrics We measure the following quantities. (i) The *optimality gap or cumulative regret* after T iterations, $\texttt{Regret}_t \doteq \sum_{t \leq T} V^*_\rho - V^{\pi^t}_\rho$. The relative difference in optimality gap between the `PMD` baseline and `PMD(+mom)` (henceforth used as shorthand for `Lazy PMD(+momentum)`) shows whether functional acceleration speeds up convergence. To quantify the complexity of the optimization problem and ill-conditioning of the optimization landscape (significant difference in scaling along different directions) for a `Random MDP` instance, we use the dual representation form of Wang et al. (2008) for policies, aka the successor representation (Dayan, 1993) or state-visitation frequency. Specifically, we define the matrix $\Psi^\pi \doteq (\mathbf{I} - \gamma \mathbf{P}^\pi)^{-1}$, with $\mathbf{P}^\pi V_s = \mathbb{E}_\pi[V_{s'}|s]$. Policy iteration is known to be equivalent to the Newton-Kantorovich iteration procedure applied to the functional equation of dynamic programming (Puterman and Brumelle, 1979), $V^{\pi^{t+1}} = V^{\pi^t} - \Psi \nabla f(V^{\pi^t})$, where $\nabla f(V) = (I - \mathcal{T})(V)$—with $\mathcal{T}$ the Bellman operator—can be treated as the gradient

operator of an unknown function $f : \mathbb{R}^{|\mathcal{S}|} \to \mathbb{R}$ (Grand-Clément, 2021) (see Appendix F). From this perspective, the matrix Ψ can be interpreted as a gradient preconditioner, its inverse is the Hessian $\nabla^2 f(V)$, the Jacobian of a gradient operator ∇f. We use the condition number of this matrix, defined as $\kappa(\Psi) \doteq {}^{|\lambda_{\max}|}/_{|\lambda_{\min}|}$, for $\lambda_{\max}$, $\lambda_{\min}$ the max and min eigenvalues in the spectrum $\text{spec}(\Psi)$. We measure (ii) the *condition number* $\kappa_0 = \kappa(\Psi^{\pi^0})$ of a randomly initialized (diffusion) policy π^0 (Fig. 1(a-b)) and (iii) the average *condition number* $\kappa_{t \leq T} = {}^1/_T \sum_{t \leq T} \kappa(\Psi^{\pi^t})$, for policies on the optimization path of an algorithm (Fig. 1(c)). Lastly, we also measure (iv) the mean *entropy* of a randomly initialized policy $\mathcal{H}_0 \propto \sum_{s,a} \pi^0_{s,a} \log \pi^0_{s,a}$ (Fig. 1(d)), inversely correlated with κ_0. Similar observations can be made using the condition number of Ψ^{-1} or the spectral radius $\rho(\Psi)$.

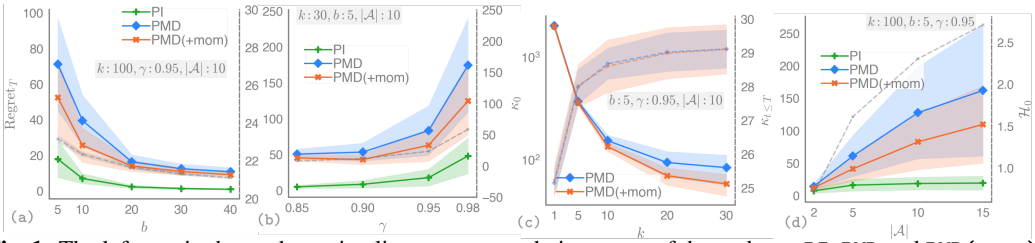

Fig. 1: The left y-axis shows the optimality gap or cumulative regret of the updates: `PI`, `PMD` and `PMD(+mom)`, after T iterations ($T = 10$ `(a-c)`, $T = 20$ `(d)`) relative to changing the hyperparameters: (a) b—the branching factor of the `Random MDP`, (b) γ—the discount factor, (c) k—the number of parameter updates, (d) $|\mathcal{A}|$—the number of actions. Shades denote standard deviation over 50 sampled MDPs. The right y-axis and dotted curves measure: `(a-b)`—the condition number κ_0, `(c)` the average condition number $\kappa_{t \leq T}$, `(d)` the entropy $\mathcal{H}_0$.

Hypothesis & Observations In Fig. 1 we show the relative difference in optimality gap ($\texttt{Regret}_t$) between `PMD(+mom)` and the `PMD` baseline, as we change the features of the algorithms and the complexity of the problem. First, we highlight two cases that lead to ill-conditioning—indicated by the condition number κ_0: `(a)` sparse connectivity of the underlying Markov chain controlled by decreasing the branching factor b, which represents the proportion of next states available at every state-action pair; `(b)` increasing the effective horizon via the discount factor γ. We illustrate the relative difference in optimality gap between the two updates correlates with ill-conditioned policy optimization landscapes, supporting the hypothesis that functional acceleration leads to faster navigation, relative to the baseline, particularly on such landscapes characterized by "long ravines" or "sharp curvatures" at the functional level. In `(c)`, we show the relative difference in optimality gap between the two updates correlates also with the magnitude of the directional derivative, captured via the number of parameter updates k used in the "inner-loop" optimization procedure. Intuitively, as $k \to \infty$ it will approach the exact solutions of the surrogate models from Sec. 4. As k decreases, the added momentum will shrink too, becoming negligible, defaulting to the classic parameter-level momentum in the limiting case of $k = 1$ (the online setting). In `(d)`, as we increase the number of actions, the optimization problem becomes more challenging, as indicated by the increasing entropy of policies and overall suboptimality. However, we also observe the relative difference between `PMD(+mom)` and the baseline `PMD` increases, suggesting the increasing advantage of functional acceleration. Appendix H.1 illustrates additional statistics on the learning performance.

Implications These studies indicate (i) that it is possible to accelerate PMD, that the advantage of functional acceleration is proportional to: (ii) the policy improvement magnitude, and (iii) the ill-conditioning of the optimization surface, induced by the policy and MDP dynamics.

5.2 Policy Dynamics in Value Space

We study the map $\pi \to V^\pi$ from stationary policies to their respective value functions. This functional mapping from policies to values has been characterized theoretically as a possibly self-intersecting, non-convex polytope (Dadashi et al., 2019). Specifically, we illustrate the expected dynamics of the functional acceleration algorithms introduced in Sec. 4 (summarized in Alg. 2 in Appendix C), over the joint simplex describing all policies. The space of value functions $\mathcal{V}$ is the set of all value functions that are attained by some policy and corresponds to the image of Π under the functional mapping $\pi \to V^\pi$: $\mathcal{V} \doteq \{V^\pi | \pi \in \Pi\}$.

Experimental Setting We use two-state MDPs (see Appendix G.1 for specifics and Appendix H.2 for additional illustrations on other MDPs). We initialize all methods at the same starting policies

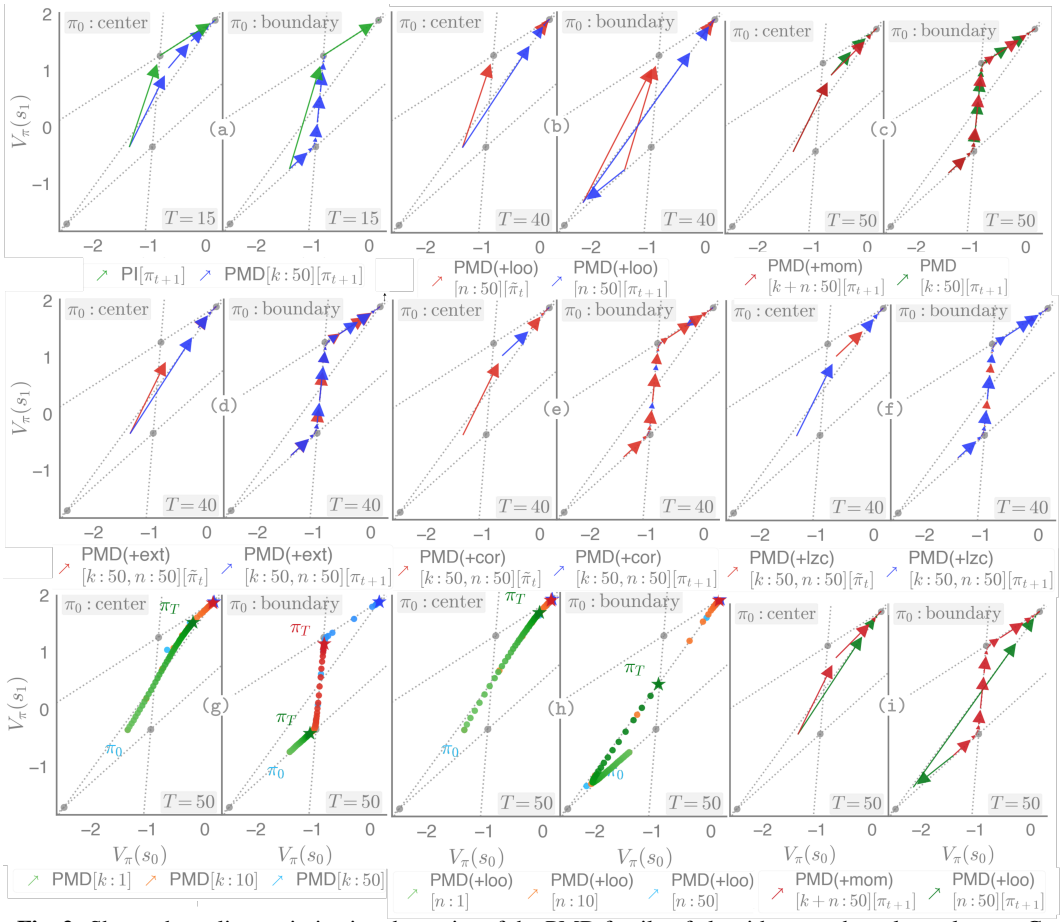

Fig. 2: Shows the policy optimization dynamics of the PMD family of algorithms on the value polytope. Gray points denote the boundaries—corresponding in this case to deterministic value functions, gray dotted lines are associated hyperplanes. `(a-f,i)` Arrows denote policy improvement between consecutive policies on the optimization path. `(g-h)` Color points denote values associated with policies on the path, color gradient indicates iteration number t, star $\star$ marks the value of the final policy π^T. Top-left annotation indicates the policy initialization, and bottom-right the final iteration number T of the snapshot.

π^0: (i) `center`—in the interior of the polytope, (ii) `boundary`—near a boundary of the polytope, close to the adversarial corner relative to the optimum. We use the value polytope to visualize three aspects of the learning dynamics: (1) the policy improvement path through the polytope, (2) the speed at which they traverse the polytope, and (3) sub-optimal attractors with long escape times that occur along this path, making the policy iterates accumulate (cf. Mei et al. (2020a)). We compute model-based versions of all relevant updates. We keep the policies $\tilde{\pi}_{\mathbf{w}}$ and π_θ separately parametrized when we compare the policy optimization dynamics of updates using two sets of policy iterates. The optimization procedure for $\mathbf{w}$ is analogous to θ using n updates. We use $\beta = 0.1$ for the "inner" loop optimization procedure and shorthand notation `PMD(+loo)`, `PMD(+ext)`, `PMD(+cor)`, `PMD(+lzc)`.

Observations & Insights Fig. 2 illustrates `(a-f)` the policy dynamics of all algorithms for a default value of $k = 50$ corresponding to an update close to being exact, `(g-h)` the impact of policy approximation through k, and `(c,i)` the benefit of approximate functional acceleration relative to the baselines: `PMD` (without acceleration) and `PMD(+loo)` (idealized acceleration).

We make the following observations: `(a)` `PMD`'s dynamics follow a straight path between the iterates of `PI`, consistent with the former being an approximation of the latter. In particular, we may infer from the accumulation points in `(g)` the dependence of the convergence speed on the approximation quality through k—the number of parameter updates per iteration. `PMD` with $k = 1$ corresponds to online PG, which depends strongly on initialization, a known issue caused by vanishing gradients at the boundary of the polytope, as we may observe in `(g)` for the π^0 : `boundary` initialization. In contrast, as we increase k, the steps become larger, and the rate of convergence is higher, revealed

by the accumulation points and the faster escape from sub-optimal attractors—the corners of the polytope. Increasing $k \to \infty$, PMD resembles PI, for which the policy iterates jump between values of deterministic policies, corresponding, in this MDP, to the corners of the polytope (a).

(b) PMD(+loo) follows a different trajectory through the value polytope compared to PMD and PI. Notably, for the π^0 : boundary, initialization, the optimization path starts in the opposite direction, guided by the lookahead (left-bottom corner) and then course-corrects. (d) PMD(+ext) accelerates toward the optimum relative to PMD. The speed of PMD(+cor) is very similar to PMD(+ext), but the optimization dynamics differ (d-e). An analogous statement applies to PMD(+lzc) vs PMD(+mom) with respect to speed similarity and dynamics difference (c,f), and correspondingly we observe acceleration for PMD(+mom) relative to PMD in (c), and suboptimality relative to the idealized acceleration of PMD(+loo) in (i).

5.3 Functional Acceleration with an Inexact Critic

For the same experimental setting as Sec. 5.2, Fig. 3 illustrates the impact of an inexact critic on the relative advantage of functional acceleration, in two settings: (Left) controlled—the critic's error is sampled from a random normal distribution with mean 0 and standard deviation τ, such that $\widehat{Q}_s^t = Q_s^t + \mathcal{N}(0, \tau), \forall s$. (Right) natural—the critic is an empirical estimate of the return obtained by Monte-Carlo sampling, and its error arises naturally from using m truncated trajectories up to horizon $1/1-\gamma$, i.e. $\widehat{Q}_s^t \doteq 1/m \sum_{i \leq m} G_s^i / N_s^i$, where G_s^i is the i^{th} empirical return sampled with π_s^t and N_s^i is the empirical visitation frequency of s.

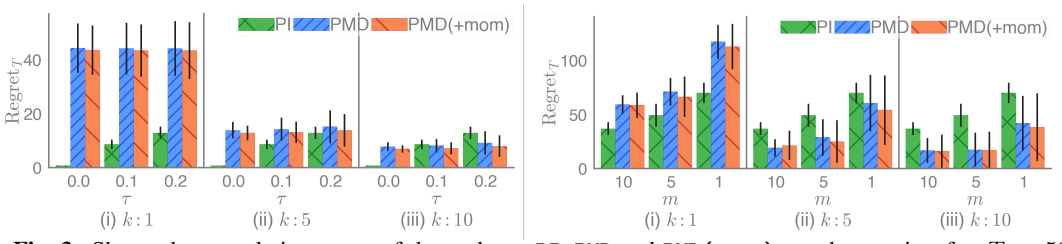

Fig. 3: Shows the cumulative regret of the updates, PI, PMD and PMD(+mom), on the y-axis, after $T = 50$ iterations, relative to changing the hyperparameter k—the number of parameter updates for PMD and PMD(+mom), with $n = 0$, in the inexact setting: (Left) controlled—τ, the scale of the critic's error, and (Right) natural—m, the number of trajectories used in the Monte-Carlo estimation of the return. Error bars denote standard deviation over 50 seeds using policies initialized from a random uniform distribution $\mathcal{U}(0, 1)$.

We observe a larger relative difference in suboptimality between PMD(+mom) and PMD for higher values of k, highlighting the difference between functional acceleration (cf. Sec.4) and classic acceleration (applied to the parameter vector θ), corresponding to $k = 1$, reinforcing evidence from Sec. 5.1. Further, we confirm PI performs increasingly poor when paired with an inexact critic with growing error. Then, we observe a range in which functional acceleration is particularly advantageous, which extends from having negligible benefit, for small k, to more impactful differences in optimality gap for larger k. Beyond a certain sweet spot, when it is maximally advantageous, the critic's error becomes too large, leading to oscillations and considerable variance. Additional illustrations of this phenomenon in Appendix H.3.

6 Closing

Inspired by functional acceleration from convex optimization theory, we proposed a momentum-based PMD update applicable to general policy parametrization and large-scale optimization. We analyzed several design choices in ablation studies designed to characterize qualitatively the properties of the resulting algorithms, and illustrated numerically how the characteristics of the problem influence the added benefit of using acceleration. Finally we looked at how inexact critics impact the method. Further analysis with these methods using stochastic simulation and function approximation would be very useful.

Acknowledgments and Disclosure of Funding

The authors thank Jincheng Mei, Hado van Hasselt and all our reviewers for feedback and insights. Veronica Chelu is grateful for support from IVADO, Fonds d'excellence en recherche Apogée Canada, Bourse d'excellence au doctorat.

References

Abdolmaleki, A., Springenberg, J. T., Tassa, Y., Munos, R., Heess, N., and Riedmiller, M. A. (2018). 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.

Agarwal, A., Kakade, S. M., Lee, J. D., and Mahajan, G. (2019). Optimality and approximation with policy gradient methods in markov decision processes. *CoRR*, abs/1908.00261.

Alfano, C., Yuan, R., and Rebeschini, P. (2024). A novel framework for policy mirror descent with general parameterization and linear convergence.

Amari, S.-i. (1998). Natural Gradient Works Efficiently in Learning. *Neural Computation*, 10(2):251–276.

Amari, S.-i. (2016). *Information Geometry and Its Applications*. Springer Publishing Company, Incorporated, 1st edition.

Anderson, D. G. M. (1965). Iterative procedures for nonlinear integral equations. *J. ACM*, 12(4):547–560.

Archibald, T. W., McKinnon, K. I. M., and Thomas, L. C. (1995). On the generation of markov decision processes. 46(3):354–361.

Banerjee, A., Merugu, S., Dhillon, I. S., and Ghosh, J. (2005). Clustering with bregman divergences. *J. Mach. Learn. Res.*, 6:1705–1749.

Beck, A. and Teboulle, M. (2003). Mirror descent and nonlinear projected subgradient methods for convex optimization. *Oper. Res. Lett.*, 31(3):167–175.

Bellemare, M. G., Dabney, W., and Munos, R. (2017). A distributional perspective on reinforcement learning. *CoRR*, abs/1707.06887.

Bellman, R. (1957). *Dynamic Programming*. Dover Publications.

Bhandari, J. and Russo, D. (2019). Global optimality guarantees for policy gradient methods. *CoRR*, abs/1906.01786.

Bhandari, J. and Russo, D. (2021). On the linear convergence of policy gradient methods for finite mdps. In Banerjee, A. and Fukumizu, K., editors, *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*, volume 130 of *Proceedings of Machine Learning Research*, pages 2386–2394. PMLR.

Bhatnagar, S., Sutton, R. S., Ghavamzadeh, M., and Lee, M. (2009). Natural actor–critic algorithms. *Automatica*, 45(11):2471–2482.

Bubeck, S. (2015). Convex optimization: Algorithms and complexity. *Found. Trends Mach. Learn.*, 8(3-4):231–357.

Böhm, A., Sedlmayer, M., Csetnek, E. R., and Boţ, R. I. (2020). Two steps at a time – taking gan training in stride with tseng's method.

Chen, G. and Teboulle, M. (1993). Convergence analysis of a proximal-like minimization algorithm using bregman functions. *SIAM J. on Optimization*, 3(3):538–543.

Cheng, C., Yan, X., Ratliff, N. D., and Boots, B. (2018). Predictor-corrector policy optimization. *CoRR*, abs/1810.06509.

Dadashi, R., Taïga, A. A., Roux, N. L., Schuurmans, D., and Bellemare, M. G. (2019). The value function polytope in reinforcement learning.

Dayan, P. (1993). Improving Generalization for Temporal Difference Learning: The Successor Representation. *Neural Computation*, 5(4):613–624.

Farahmand, A.-M. and Ghavamzadeh, M. (2021). Pid accelerated value iteration algorithm. In Meila, M. and Zhang, T., editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 3143–3153. PMLR.

Gidel, G., Berard, H., Vincent, P., and Lacoste-Julien, S. (2018). A variational inequality perspective on generative adversarial nets. *CoRR*, abs/1802.10551.

Goyal, V. and Grand-Clement, J. (2021). A first-order approach to accelerated value iteration.

Grand-Clément, J. (2021). From convex optimization to mdps: A review of first-order, second-order and quasi-newton methods for mdps.

Haarnoja, T., Zhou, A., Abbeel, P., and Levine, S. (2018). Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In Dy, J. G. and Krause, A., editors, *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 1856–1865. PMLR.

Hessel, M., Modayil, J., van Hasselt, H., Schaul, T., Ostrovski, G., Dabney, W., Horgan, D., Piot, B., Azar, M. G., and Silver, D. (2017). Rainbow: Combining improvements in deep reinforcement learning. *CoRR*, abs/1710.02298.

Hinton, G., Srivastava, N., and Swersky, K. (2012). Neural networks for machine learning lecture 6a overview of mini-batch gradient descent. *CSC321*.

Johnson, E., Pike-Burke, C., and Rebeschini, P. (2023). Optimal convergence rate for exact policy mirror descent in discounted markov decision processes.

Joulani, P., György, A., and Szepesvári, C. (2020a). A modular analysis of adaptive (non-)convex optimization: Optimism, composite objectives, variance reduction, and variational bounds. *Theor. Comput. Sci.*, 808:108–138.

Joulani, P., Raj, A., Gyorgy, A., and Szepesvari, C. (2020b). A simpler approach to accelerated optimization: iterative averaging meets optimism. In III, H. D. and Singh, A., editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 4984–4993. PMLR.

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

Jumper, J., Evans, R., Pritzel, A., Green, T., Figurnov, M., Ronneberger, O., Tunyasuvunakool, K., Bates, R., Žídek, A., Potapenko, A., Bridgland, A., Meyer, C., Kohl, S. A. A., Ballard, A. J., Cowie, A., Romera-Paredes, B., Nikolov, S., Jain, R., Adler, J., Back, T., Petersen, S., Reiman, D., Clancy, E., Zielinski, M., Steinegger, M., Pacholska, M., Berghammer, T., Bodenstein, S., Silver, D., Vinyals, O., Senior, A. W., Kavukcuoglu, K., Kohli, P., and Hassabis, D. (2021). Highly accurate protein structure prediction with alphafold. *Nature*, 596(7873):583–589.

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

Kakade, S. M. (2001). A natural policy gradient. In Dietterich, T., Becker, S., and Ghahramani, Z., editors, *Advances in Neural Information Processing Systems*, volume 14. MIT Press.

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

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

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

Lu, H., Freund, R. M., and Nesterov, Y. (2017). Relatively-smooth convex optimization by first-order methods, and applications.

Malitsky, Y. and Tam, M. K. (2020). A forward-backward splitting method for monotone inclusions without cocoercivity.

Mei, J., Dai, B., Xiao, C., Szepesvári, C., and Schuurmans, D. (2021). Understanding the effect of stochasticity in policy optimization. *CoRR*, abs/2110.15572.

Mei, J., Xiao, C., Dai, B., Li, L., Szepesvári, C., and Schuurmans, D. (2020a). Escaping the gravitational pull of softmax. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H., editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Mei, J., Xiao, C., Szepesvári, C., and Schuurmans, D. (2020b). On the global convergence rates of softmax policy gradient methods. *CoRR*, abs/2005.06392.

Mnih, V., Badia, A. P., Mirza, M., Graves, A., Lillicrap, T. P., Harley, T., Silver, D., and Kavukcuoglu, K. (2016). Asynchronous methods for deep reinforcement learning. *CoRR*, abs/1602.01783.

Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., and Riedmiller, M. A. (2013). Playing atari with deep reinforcement learning. *CoRR*, abs/1312.5602.

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

Nemirovski, A. (2004). 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.

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

Nie, X., Brunskill, E., and Wager, S. (2020). Learning when-to-treat policies.

Parikh, N., Boyd, S., et al. (2014). Proximal algorithms. *Foundations and trends® in Optimization*, 1(3):127–239.

Puterman, M. L. (1994). *Markov Decision Processes*. Wiley.

Puterman, M. L. and Brumelle, S. L. (1979). On the convergence of policy iteration in stationary dynamic programming. *Mathematics of Operations Research*, 4(1):60–69.

Rakhlin, A. and Sridharan, K. (2013). Online learning with predictable sequences. In Shalev-Shwartz, S. and Steinwart, I., 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.

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

Russo, D. (2022). Approximation benefits of policy gradient methods with aggregated states.

Schaefer, A. J., Bailey, M. D., Shechter, S. M., and Roberts, M. S. (2004). *Modeling Medical Treatment Using Markov Decision Processes*, pages 593–612. Springer US, Boston, MA.

Scherrer, B. and Geist, M. (2014). Local policy search in a convex space and conservative policy iteration as boosted policy search. In *Machine Learning and Knowledge Discovery in Databases*, page 35–50, Berlin, Heidelberg. Springer-Verlag.

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

Schulman, J., Levine, S., Moritz, P., Jordan, M. I., and Abbeel, P. (2015). Trust region policy optimization. *CoRR*, abs/1502.05477.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. (2017). Proximal policy optimization algorithms. *CoRR*, abs/1707.06347.

Shannon, C. E. (1948). A mathematical theory of communication. *The Bell system technical journal*, 27(3):379–423.

Silver, D., Lever, G., Heess, N., Degris, T., Wierstra, D., and Riedmiller, M. (2014). Deterministic policy gradient algorithms. In *Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32*, ICML'14, page I–387–I–395. JMLR.org.

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

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

Sutton, R. S., McAllester, D., Singh, S., and Mansour, Y. (1999). 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. MIT Press.

Tesauro, G. (1994). TD-Gammon, a Self-Teaching Backgammon Program, Achieves Master-Level Play. *Neural Computation*, 6(2):215–219.

Tomar, M., Shani, L., Efroni, Y., and Ghavamzadeh, M. (2020). Mirror descent policy optimization. *CoRR*, abs/2005.09814.

Tseng, P. (1991). Applications of a splitting algorithm to decomposition in convex programming and variational inequalities. *SIAM Journal on Control and Optimization*, 29(1):119–138.

Vaswani, S., Bachem, O., Totaro, S., Mueller, R., Geist, M., Machado, M. C., Castro, P. S., and Roux, N. L. (2021). A functional mirror ascent view of policy gradient methods with function approximation. *CoRR*, abs/2108.05828.

Vaswani, S., Kazemi, A., Babanezhad, R., and Roux, N. L. (2023). Decision-aware actor-critic with function approximation and theoretical guarantees.

Vieillard, N., Scherrer, B., Pietquin, O., and Geist, M. (2019). Momentum in reinforcement learning. *CoRR*, abs/1910.09322.

Wang, J. and Abernethy, J. D. (2018). Acceleration through optimistic no-regret dynamics. *CoRR*, abs/1807.10455.

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

Wang, T., Lizotte, D. J., Bowling, M., and Schuurmans, D. (2008). Dual representations for dynamic programming.

Williams, R. J. (1992). Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Mach. Learn.*, 8:229–256.

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

Yuan, R., Du, S. S., Gower, R. M., Lazaric, A., and Xiao, L. (2023). Linear convergence of natural policy gradient methods with log-linear policies. In *The Eleventh International Conference on Learning Representations*.

Zahavy, T., Veeriah, V., Hou, S., Waugh, K., Lai, M., Leurent, E., Tomasev, N., Schut, L., Hassabis, D., and Singh, S. (2023). Diversifying ai: Towards creative chess with alphazero.

A Notation

t	iteration number
T	max number of iterations
k, n	number of GD updates for the "inner-loop" proximal optimization procedure
$\eta, \tilde{\eta}$	step sizes for the proximal update (regularization strength of the divergence)
β	step size for the "inner-loop" parameter-level optimization procedure
h	the mirror map
$D_h(\pi, \tilde{\pi})$	Bregman divergence associated with the mirror map h

Table 1: Notation

B Proofs and derivations

B.1 Proofs and Derivations for Sec.3: Background & Preliminaries

B.1.1 Functional Policy Gradient

The Performance Difference Lemma (PDL) is a property that relates the difference in values of policies to the policies themselves.

Lemma 1. ***(Performance Difference Lemma from Kakade and Langford (2002))*** *For any policies π^{t+1} and π^t, and an initial distribution ρ*

$$V_\rho^{t+1} - V_\rho^t = {}^1\!/_{1-\gamma} \sum_s \sum_{a \in \mathcal{A}} d_{s,\rho}^{t+1}(\pi_{s,a}^{t+1} - \pi_{s,a}^t, Q_{s,a}^t) = {}^1\!/_{1-\gamma} \mathbb{E}_{s \sim d_\rho^{t+1}} \left[\langle Q_s^t, \pi_s^{t+1} - \pi_s^t \rangle \right]$$

Proof. According to the definition of the value function

$$\begin{aligned} V_s^{t+1} - V_s^t &= \langle Q_s^{t+1}, \pi_s^{t+1} \rangle - \langle Q_s^t, \pi_s^t \rangle \\ &= \langle Q_s^t, \pi_s^{t+1} - \pi_s^t \rangle + \langle Q_s^{t+1} - Q_s^t, \pi_s^{t+1} \rangle \\ &= \langle Q_s^t, \pi_s^{t+1} - \pi_s^t \rangle + \gamma \sum_{s'} \sum_a P(s'|s,a) \pi_{s,a}^{t+1} [V_{s'}^{t+1} - V_{s'}^t] \\ &= {}^1\!/_{1-\gamma} \sum_{s'} d_{s \to s'}^{t+1} \langle Q_{s'}^t, \pi_{s'}^{t+1} - \pi_{s'}^t \rangle \end{aligned}$$

□

The following lemma is a version of the policy gradient theorem (Sutton et al., 1999) applied to the direct policy representation—the functional representation of the policy probabilities, and has appeared in various forms in Agarwal et al. (2019); Bhandari and Russo (2019, 2021); Russo (2022).

Lemma 2. ***(Policy Gradient Theorem for Directional Derivatives)*** *For two policies $\pi^{t+1}, \pi^t \in \Pi$*

$$\langle \nabla V_\rho^t, \pi^{t+1} - \pi^t \rangle = \sum_{s \in \mathcal{S}} \sum_{a \in \mathcal{A}} d_s^{\pi^t} Q_{s,a}^t (\pi_{s,a}^{t+1} - \pi_{s,a}^t) = \mathbb{E}_{s \sim d^t} \left[\langle Q_s^t, \pi_s^{t+1} - \pi_s^t \rangle \right]$$

Proof. A Taylor expansion using the Performance Difference Lemma 1 reveals

$$\begin{aligned} V_\rho^{t+1} - V_\rho^t &= {}^1\!/_{1-\gamma} \sum_s \sum_{a \in \mathcal{A}} d_{s,\rho}^{t+1}(\pi_{s,a}^{t+1} - \pi_{s,a}^t, Q_{s,a}^t) = \mathbb{E}_{s \sim d_\rho^{t+1}} \left[\langle Q_s^t, \pi_s^{t+1} - \pi_s^t \rangle \right] \\ &= {}^1\!/_{1-\gamma} \mathbb{E}_{s \sim d_\rho^t} \left[\langle Q_s^t, \pi_s^{t+1} - \pi_s^t \rangle \right] + {}^1\!/_{1-\gamma} \underbrace{\sum_{s \in \mathcal{S}} (d_{s,\rho}^{t+1} - d_{s,\rho}^t) \langle Q_s^t, \pi_s^{t+1} - \pi_s^t \rangle}_{= O(\|\pi_s^{t+1} - \pi_s^t\|^2)} \end{aligned}$$

The last error term is second-order since P^π is linear in π and then d_ρ^π is differentiable in π. □

The next lemma states that the MD method minimizes the local linearization of a function while not moving too far away from the previous point, with distances measured via the Bregman divergence of the mirror map.

Lemma 3. ***(Proximal perspective on mirror descent)*** *The MD update for $x \in \mathcal{X} \cap \mathcal{C}$, with mirror map $h : \mathcal{X} \to \mathbb{R}$ for the minimization problem $\min_{x \in \mathcal{X} \cap \mathcal{C}} f(x)$, with $f : \mathcal{X} \to \mathbb{R}$ can be rewritten in the following ways, for step-size $\eta \geq 0$ and $t \geq 0$*

$$\begin{aligned} x_{t+1} &= \operatorname{argmin}_{x \in \mathcal{X} \cap \mathcal{C}} D_h(x, \nabla h^*(\nabla h(x_t) + \eta \nabla f(x_t))) && \textit{(PGD)} \\ &= \operatorname{argmin}_{x \in \mathcal{X} \cap \mathcal{C}} \eta \langle \nabla f(x_t), x \rangle + D_h(x, x_t) && \textit{(proximal perspective)} \end{aligned}$$

Proof.

$$\begin{aligned} x_{t+1} &= \operatorname{argmin}_{x \in \mathcal{X} \cap \mathcal{C}} D_h(x, \nabla h^*(\nabla h(x_t) + \eta \nabla f(x_t))) && \text{(generalized GD)} \\ &= \operatorname{argmin}_{x \in \mathcal{X} \cap \mathcal{C}} h(x) - \langle \nabla h(\nabla h^*(\nabla h(x_t) + \eta \nabla f(x_t))), x \rangle \\ &= \operatorname{argmin}_{x \in \mathcal{X} \cap \mathcal{C}} h(x) - \langle \nabla h(x_t) + \eta \nabla f(x_t), x \rangle \\ &= \operatorname{argmin}_{x \in \mathcal{X} \cap \mathcal{C}} \eta \langle \nabla f(x_t), x \rangle + D_h(x, x_t) && \text{(proximal perspective)} \end{aligned}$$

□

B.1.2 Helpful Lemmas for Policy Mirror Descent

Key to the analysis of Xiao (2022) and Johnson et al. (2023) is the Three-Point Descent Lemma, that relates the improvement of the proximal gradient update compared to an arbitrary point. It originally comes from Chen and Teboulle (1993) (Lemma 3.2).

Lemma 4. ***(Three-Point Descent Lemma, Lemma 6 in Xiao (2022))***. *Suppose that $\mathcal{X} \subset \mathbb{R}^n$ is a closed convex set, $\psi : \mathcal{X} \to \mathbb{R}$ is a proper, closed convex function, $D_h(\cdot, \cdot)$ is the Bregman divergence generated by a function h of Legendre type and $\operatorname{rint} \operatorname{dom} h \cap \mathcal{X} \neq \emptyset$. For any $x^t \in \operatorname{rint} \operatorname{dom} h$, let*

$$x^{t+1} = \operatorname{argmin}_{x \in \mathcal{X}} \psi(x) + D_h(x, x^t)$$

Then $x^{t+1} \in \operatorname{rint} \operatorname{dom} h \cap \mathcal{X}$ and $\forall x \in \mathcal{X}$,

$$\psi(x^{t+1}) + D_h(x^{t+1}, x^t) \leq \psi(x) + D_h(x, x^t) - D_h(x, x^{t+1})$$

The PMD update is an instance of the proximal minimisation with $\mathcal{X} = \Delta(\mathcal{A}), x^t = \pi_s^t$ and $\psi(x) = -\eta^t \langle Q_s^t, x \rangle$. Plugging these in, the Three-Point Descent Lemma relates the decrease in the proximal objective of π_s^{t+1} to any other policy, i.e. $\forall \pi_s \in \Delta(\mathcal{A})$,

$$-\eta^t \langle Q_s^t, \pi_s^{t+1} - \pi_s \rangle \leq D_h(\pi_s, \pi_s^t) - D_h(\pi_s^{t+1}, \pi_s^t) - D_h(\pi_s, \pi_s^{t+1})$$

This equation is key to the analysis of convergence of exact PMD, leading to Lemma 6 regarding the monotonic improvement in Q-functions of PMD iterates.

Lemma 5. ***(Descent Property of PMD for Q-functions, Lemma 7 in Xiao (2022)*** *Consider the policies produced by the iterative updates of exact PMD. For any $t \geq 0$*

$$\langle Q_s^t, \pi_s^{t+1} - \pi_s^t \rangle \geq 0, \quad \forall s \in \mathcal{S},$$

Proof. From the Three-Point Descent Lemma 4 of Xiao (2022) with $\pi_s = \pi_s^t$,

$$\eta^t \langle Q_s^t, \pi_s^{t+1} - \pi_s^t \rangle \geq D_h(\pi_s^t, \pi_s^{t+1}) + D_h(\pi_s^{t+1}, \pi_s^t)$$

since the Bregman divergences are non-negative and $\eta^t > 0$,

$$\langle Q_s^t, \pi_s^{t+1} - \pi_s^t \rangle \geq 0$$

□

Lemma 6. ***(Descent Property of PMD for Value Functions, Lemma A.2. from Johnson et al. (2023))*** *Consider the policies produced by the iterative updates of exact PMD. Then for any $t \geq 0$,*

$$\begin{aligned} Q_s^{t+1} &\geq Q_s^t, \quad \forall s \in \mathcal{S} \\ V_\rho^{t+1} &\geq V_\rho^t, \quad \forall \rho \in \Delta(\mathcal{S}) \end{aligned}$$

Proof. Follows from Lemma 5 by an application of the Performance Difference Lemma 1, for an initial state distribution ρ

$$V_\rho^{t+1} - V_\rho^t = \frac{1}{1-\gamma} \mathbb{E}_{s \sim d_\rho^{t+1}} \left[\langle Q_s^t, \pi_s^{t+1} - \pi_s^t \rangle \right] \geq 0$$

□

B.1.3 Detailed Derivation of the Suboptimality Decomposition and Convergence of PMD

Suboptimality decomposition Fix a state s. For any $\pi_s, \tilde{\pi}_s$, let $D_{-V}(\pi_s, \tilde{\pi}_s)$ be analogous to a standard Bregman divergence with mirror map $-V$, capturing the curvature of $-V$ at π_s

$$\begin{aligned} D_{-V}(\pi_s, \tilde{\pi}_s) &\doteq -V_s^{\pi_s} - (-V_s^{\tilde{\pi}_s}) - \langle -Q_s^{\tilde{\pi}_s}, \pi_s - \tilde{\pi}_s \rangle \\ &\doteq -V_s^{\pi_s} + V_s^{\tilde{\pi}_s} + \langle Q_s^{\tilde{\pi}_s}, \pi_s - \tilde{\pi}_s \rangle \\ &= -\langle Q_s^{\pi} - Q_s^{\tilde{\pi}}, \pi_s \rangle \qquad (8) \\ &\text{(using Holder's inequality)} \qquad (9) \\ &\geq -\|Q_s^{\pi} - Q_s^{\tilde{\pi}}\|_\infty \|\pi_s\|_1 \\ &\geq -\gamma \|V^{\pi} - V^{\tilde{\pi}}\|_\infty \qquad (10) \end{aligned}$$

For the general case, using the approximation $\widehat{Q}_s^t \approx Q_s^t$, the per-iteration suboptimality is

$$V_s^* - V_s^t = -\langle \widehat{Q}_s^t, \pi_s^{t+1} - \pi_s^* \rangle - \langle \widehat{Q}_s^t, \pi_s^t - \pi_s^{t+1} \rangle - D_{-V}(\pi_s^*, \pi_s^t) - \langle Q_s^t - \widehat{Q}_s^t, \pi_s^t - \pi_s^* \rangle \quad (11)$$

The first term, $-\langle \widehat{Q}_s^t, \pi_s^{t+1} - \pi_s^* \rangle$, is the forward regret (cf. Joulani et al. (2020a)), defined as the regret of a "cheating" algorithm that uses the π^{t+1} at time t, and depends only on the choices of the algorithm and the feedback it receives. This quantity can be can be upper-bounded using an idealized lookahead policy, $\bar{\pi}_s^{t+1}$—greedy with respect to $\widehat{Q}_s^t$ (cf. Johnson et al. (2023)).

If π_s^{t+1} is the result of a PMD update, then Johnson et al. (2023) show that using the Three-Point Descent Lemma (Lemma 6, Xiao (2022), included in Appendix B.1, Lemma 4), denoting the step sizes $\eta^t \geq 0$, the forward regret is further upper-bounded by

$$\begin{aligned} -\langle \widehat{Q}_s^t, \pi_s^{t+1} - \pi_s^* \rangle &\leq -\langle \widehat{Q}_s^t, \pi^{t+1} - \bar{\pi}^{t+1} \rangle \\ &\leq \langle \widehat{Q}_s^t, \bar{\pi}^{t+1} - \pi^{t+1} \rangle \\ &\leq {}^1\!/\!_{\eta^t} D_h(\bar{\pi}_s^{t+1}, \pi_s^t) - {}^1\!/\!_{\eta^t} D_h(\bar{\pi}_s^{t+1}, \pi_s^{t+1}) - {}^1\!/\!_{\eta^t} D_h(\pi_s^{t+1}, \pi_s^t) \\ &\leq {}^1\!/\!_{\eta^t} D_h(\bar{\pi}_s^{t+1}, \pi_s^t) \qquad (12) \end{aligned}$$

The second term in Eq. 11 is

$$-\langle \widehat{Q}_s^t, \pi_s^t - \pi_s^{t+1} \rangle = V_s^{t+1} - V_s^t + D_{-V}(\pi_s^{t+1}, \pi_s^t) + \langle Q_s^t - \widehat{Q}_s^t, \pi_s^t - \pi_s^{t+1} \rangle \quad (13)$$

The third term $-D_{-V}(\pi_s^*, \pi_s^t)$ can be bounded by applying the upper-approximation from Eq.10, resulting in

$$-D_{-V}(\pi_s^*, \pi_s^t) \leq \gamma \|V^* - V^t\|_\infty \quad (14)$$

Plugging Eq. 12, 13, 14 back in the suboptimality decomposition from Eq. 11, we obtain

$$V_s^* - V_s^{t+1} \leq \gamma \|V^* - V^t\|_\infty + \underbrace{{}^1\!/\!_{\eta^t} D_h(\bar{\pi}_s^{t+1}, \pi_s^t) + \langle Q_s^t - \widehat{Q}_s^t, \pi_s^* - \pi_s^{t+1} \rangle + D_{-V}(\pi_s^{t+1}, \pi_s^t)}_{\xi_t \text{(iteration error)}} \quad (15)$$

With ξ_t—the iteration error, recursing Eq. 15

$$\|V^* - V^t\|_\infty \leq \gamma^t \textstyle\sum_{i \leq t} \xi_i / \gamma^i + \gamma^t \|V^* - V^0\|_\infty$$

Convergence of Exact PMD at γ-rate If the PMD update is exact, then $\widehat{Q}_s^t = Q_s^t, \forall s \in \mathcal{S}$. The Three-Point Descent Lemma 4 guarantees policy improvement for an Exact PMD update, and yields Lemma 6 stating $V_s^{t+1} \geq V_s^t$, and $\langle Q_s^{t+1} - Q_s^t, \pi_s^{t+1} \rangle \geq 0$. Consequently

$$D_{-V}(\pi_s^{t+1}, \pi_s^t) = -\langle Q_s^{t+1} - Q_s^t, \pi_s^{t+1} \rangle \leq 0$$

There remains only one term in the suboptimality from Eq. 11, namely

$$\xi_t \leq {}^1\!/\!_{\eta^t} D_h(\bar{\pi}_s^{t+1}, \pi_s^t)$$

An optimal step-size η^t can be derived by upper-bounding it ${}^1\!/\!_{\eta^t} D_h(\bar{\pi}_s^{t+1}, \pi_s^t) \leq \epsilon_t$, for any arbitrary constant ϵ_t. Setting $\epsilon_t = \gamma^{2(t+1)} \epsilon_0$ for some $\epsilon_0 > 0$, gives the optimal step-size with a geometrically increasing component, which guarantees linear convergence at the γ-rate

$$\|V^* - V^t\|_\infty \leq \gamma^t \left(\left\| V^* - V^0 \right\|_\infty + {}^{\epsilon_0}\!/\!_{1-\gamma} \right)$$

matching the bounds of PI and VI as ϵ_0 goes to 0 (cf. Theorem 4.1., Johnson et al. (2023)).

B.2 Proofs for Sec. 4: Functional Acceleration for PMD

Definition 7. (Functional gradient of the Bregman divergence) Fix a state s. For any policies π_s^1, π_s^0, we denote the gradient of the Bregman divergence with respect to the first argument

$$\nabla D_h(\pi_s^1, \pi_s^0) \doteq \nabla h(\pi_s^1) - \nabla h(\pi_s^0)$$

The following lemma can be also interpreted as a definition for the difference of differences of Bregman divergences.

Lemma 8. ***(Four-Point Identity Lemma of Bregman divergences)*** *For any four policies $\pi^3, \pi^2, \pi^1, \pi^0$, we have*

$$\langle \nabla D_h(\pi_s^1, \pi_s^0), \pi_s^3 - \pi_s^2 \rangle = D_h(\pi_s^3, \pi_s^0) - D_h(\pi_s^3, \pi_s^1) - [D_h(\pi_s^2, \pi_s^0) - D_h(\pi_s^2, \pi_s^1)]$$

Proof. Immediate from the definition. □

An immediate consequence is the Three-Point Identity Lemma of Bregman divergences (cf. Bubeck (2015), Eq. 4.1, Beck and Teboulle (2003), Lemma 4.1).

Lemma 9. ***(Three-Point Identity Lemma of Bregman divergences)*** *For any three policies π^2, π^1, π^0,*

$$\langle \nabla D_h(\pi_s^1, \pi_s^0), \pi_s^2 - \pi_s^1 \rangle = D_h(\pi_s^2, \pi_s^0) - D_h(\pi_s^2, \pi_s^1) - D_h(\pi_s^1, \pi_s^0)$$

Proof. Apply Lemma 8 with $\pi_s^3 = \pi_s^2$, $\pi_s^2 = \pi_s^1$. □

Proposition 10. ***(Extrapolation from the future)*** *The solutions of* `PMD(+extragradient)` *subsume those of* `PMD(+correction)`.

Proof. Fix a state s and timestep t. From the definition of `PMD(+extragradient)`

$$\begin{aligned}
\tilde{\pi}_s^t &= \operatorname{argmin}_{\pi_s \in \Delta(\mathcal{A})} -\langle \widehat{Q}_s^t, \pi_s \rangle + {}^1\!/\!_{\eta^t} D_h(\pi_s, \pi_s^t) \\
\tilde{Q}_s^t &= \mathbb{E}[r_s + \gamma \widehat{Q}^t(s', \tilde{\pi}_s^t)]
\end{aligned}$$

given step-sizes η^t, π_s^t, $\widehat{Q}_s^t$. For some Q_s^t, η^t and $\forall \pi_s \in \Delta(\mathcal{A})$, let

$$\ell(\pi_s, \nabla h(\pi_s^t) - \eta^t Q_s^t) \doteq -\eta^t \langle Q_s^t, \pi_s \rangle + D_h(\pi_s, \pi_s^t)$$

With this notation, given step-sizes $\tilde{\eta}^t$, we write the surrogate objectives for the next policy iterates of the updates of `PMD(+extragradient)`

$$\pi_s^{t+1} \doteq \operatorname{argmin}_{\pi_s \in \Delta(\mathcal{A})} \underbrace{-\langle \tilde{Q}_s^t, \pi_s \rangle + {}^1\!/\!_{\tilde{\eta}^t} D_h(\pi_s, \pi_s^t)}_{\ell(\pi_s, \nabla h(\pi^t) - \tilde{\eta}^t \tilde{Q}_s^t)}$$

and those of `PMD(+correction)`,

$$\pi_s^{t+1} \doteq \operatorname{argmin}_{\pi_s \in \Delta(\mathcal{A})} \underbrace{-\langle \tilde{Q}_s^t - {}^{\eta^t}\!/\!_{\tilde{\eta}^t} \widehat{Q}_s^t, \pi_s \rangle + {}^1\!/\!_{\tilde{\eta}^t} D_h(\pi_s, \tilde{\pi}_s^t)}_{\ell(\pi_s, \nabla h(\tilde{\pi}_s^t) - \tilde{\eta}^t [\tilde{Q}_s^t - {}^{\eta^t}\!/\!_{\tilde{\eta}^t} \widehat{Q}_s^t])}$$

Using Lemma 9 with $\pi_s^1 = \tilde{\pi}_s^t$, $\pi_s^2 = \pi_s$ $\pi_s^0 = \pi_s^t$

$$\langle \nabla D_h(\tilde{\pi}_s^t, \pi_s^t), \pi_s - \tilde{\pi}_s^t \rangle + D_h(\pi_s, \tilde{\pi}_s^t) = D(\pi_s, \pi_s^t) - D(\tilde{\pi}_s^t, \pi_s^t) \tag{16}$$

From the optimality of $\tilde{\pi}_s^t$, using the Three-Point Descent Lemma 4, $\forall \pi_s$

$$\begin{aligned}
\langle \widehat{Q}_s^t, \pi_s - \tilde{\pi}_s^t \rangle + {}^1\!/\!_{\eta^t} D_h(\pi_s, \tilde{\pi}_s^t) &\leq {}^1\!/\!_{\eta^t} \left(\langle \nabla D_h(\tilde{\pi}_s^t, \pi_s^t), \pi_s - \tilde{\pi}_s^t \rangle + D_h(\pi_s, \tilde{\pi}_s^t) \right) \\
&= {}^1\!/\!_{\eta^t} \left(D_h(\pi_s, \pi_s^t) - D_h(\tilde{\pi}_s^t, \pi_s^t) \right) \\
\implies \forall \eta \geq 0 \quad \langle {}^{\eta^t}\!/\!_{\eta} \widehat{Q}_s^t, \pi_s - \tilde{\pi}_s^t \rangle + {}^1\!/\!_{\eta} D_h(\pi_s, \tilde{\pi}_s^t) &\leq {}^1\!/\!_{\eta} \left(\langle \nabla D_h(\tilde{\pi}_s^t, \pi_s^t), \pi_s - \tilde{\pi}_s^t \rangle + D_h(\pi_s, \tilde{\pi}_s^t) \right) \\
&= {}^1\!/\!_{\eta} \left(D_h(\pi_s, \pi_s^t) - D_h(\tilde{\pi}_s^t, \pi_s^t) \right)
\end{aligned} \tag{17}$$

Plugging Eq. 16 and Eq. 17 in the definition of the PMD(+extragradient) update, we can observe the objectives of PMD(+extragradient) and PMD(+correction) are related in the following way

$$
\begin{aligned}
&\ell(\pi_s, \nabla h(\pi^t) - \tilde{\eta}^t \tilde{Q}_s^t) - \ell(\tilde{\pi}_s^t, \nabla h(\pi^t) - \tilde{\eta}^t \tilde{Q}_s^t) \\
&= -\langle \tilde{Q}_s^t, \pi_s \rangle + {}^1\!/_{\tilde{\eta}^t} D_h(\pi_s, \pi_s^t) - \left[-\langle \tilde{Q}_s^t, \tilde{\pi}_s^t \rangle + {}^1\!/_{\tilde{\eta}^t} D_h(\tilde{\pi}_s^t, \pi_s^t) \right] \\
&= -\langle \tilde{Q}_s^t, \pi_s - \tilde{\pi}_s^t \rangle + {}^1\!/_{\tilde{\eta}^t} \left[D_h(\pi_s, \pi_s^t) - D_h(\tilde{\pi}_s^t, \pi_s^t) \right] \\
&\overset{Eq.\ 16}{=} -\langle \tilde{Q}_s^t - {}^1\!/_{\eta} \nabla D_h(\tilde{\pi}_s^t, \pi_s^t), \pi_s - \tilde{\pi}_s^t \rangle + {}^1\!/_{\tilde{\eta}^t} D_h(\pi_s, \tilde{\pi}_s^t) \\
&= \ell(\pi_s, \nabla h(\tilde{\pi}_s^t) - \tilde{\eta}^t [\tilde{Q}_s^t - {}^1\!/_{\tilde{\eta}^t} \nabla D_h(\tilde{\pi}_s^t, \pi_s^t)]) - \ell(\tilde{\pi}_s^t, \nabla h(\tilde{\pi}_s^t) - \tilde{\eta}^t [\tilde{Q}_s^t - {}^1\!/_{\tilde{\eta}^t} \nabla D_h(\tilde{\pi}_s^t, \pi_s^t)]) \\
&\overset{Eq.\ 17}{\geq} -\langle \tilde{Q}_s^t - {}^{\eta^t}\!/_{\tilde{\eta}^t} \widehat{Q}_s^t, \pi_s - \tilde{\pi}_s^t \rangle + {}^1\!/_{\tilde{\eta}^t} D_h(\pi_s, \tilde{\pi}_s^t) \\
&= \ell(\pi_s, \nabla h(\tilde{\pi}_s^t) - \tilde{\eta}^t [\tilde{Q}_s^t - {}^{\eta^t}\!/_{\tilde{\eta}^t} \widehat{Q}_s^t]) - \ell(\tilde{\pi}_s^t, \nabla h(\tilde{\pi}_s^t) - \tilde{\eta}^t [\tilde{Q}_s^t - {}^{\eta^t}\!/_{\tilde{\eta}^t} \widehat{Q}_s^t])
\end{aligned}
$$

Ignoring constant terms, the ordering over objectives implies the solutions of PMD(+extragradient) subsume those of PMD(+correction). □

Proposition 11. ***(Extrapolation from the past)*** *The solutions of* Lazy PMD(+momentum) *subsume those of* Lazy PMD(+correction).

Proof. Fix a state s and timestep t. With $\tilde{\pi}_s^t$ from the definition of Lazy PMD(+correction)

$$\tilde{\pi}_s^t = \operatorname{argmin}_{\pi_s \in \Delta(\mathcal{A})} -\langle {}^{\eta^{t-1}}\!/_{\eta_t} (\widehat{Q}_s^t - \widehat{Q}_s^{t-1}), \pi_s \rangle + {}^1\!/_{\eta^{t-1}} D_h(\pi_s, \pi_s^t)$$

given step-size η_s^t, π_s^t, $\widehat{Q}_s^t$. Given some Q_s^t, η^t, let

$$\ell(\pi_s, \nabla h(\pi_s^t) - \eta Q_s^t) \doteq -\eta^t \langle Q_s^t, \pi_s \rangle + D_h(\pi_s, \pi_s^t)$$

$\forall \pi_s \in \Delta(\mathcal{A})$.

With this notation, we may write the surrogate objectives for the next policy iterates of Lazy PMD(+correction)

$$\pi_s^{t+1} = \operatorname{argmin}_{\pi_s \in \Delta(\mathcal{A})} \underbrace{-\langle \widehat{Q}_s^t, \pi_s \rangle + {}^1\!/_{\eta^t} D_h(\pi_s, \tilde{\pi}_s^t)}_{\ell(\pi_s, \nabla h(\tilde{\pi}_s^t) - \eta^t \widehat{Q}_s)}$$

and Lazy PMD(+momentum)

$$\pi_s^{t+1} = \operatorname{argmin}_{\pi_s \in \Delta(\mathcal{A})} \underbrace{-\langle \widehat{Q}_s^t + {}^{\eta^{t-1}}\!/_{\eta^t} (\widehat{Q}_s^t - \widehat{Q}_s^{t-1}), \pi_s \rangle + {}^1\!/_{\eta^t} D_h(\pi_s, \pi_s^t)}_{\ell(\pi_s, \nabla h(\pi_s^t) - \eta^t [\widehat{Q}_s^t + {}^{\eta^{t-1}}\!/_{\eta^t} (\widehat{Q}_s^t - \widehat{Q}_s^{t-1}))}$$

Using Lemma 9 with $\pi_s^1 = \tilde{\pi}_s^t$, $\pi_s^2 = \pi_s$ $\pi_s^0 = \pi_s^t$, we have

$$\langle \nabla D_h(\tilde{\pi}_s^t, \pi_s^t), \pi_s - \tilde{\pi}_s^t \rangle + D_h(\pi_s, \tilde{\pi}_s^t) = D(\pi_s, \pi_s^t) - D(\tilde{\pi}_s^t, \pi_s^t) \tag{18}$$

From the optimality of $\tilde{\pi}_s^t$, for any π_s, using the Three-Point Descent Lemma 4

$$
\begin{aligned}
\ell(\tilde{\pi}_s^t, \nabla h(\pi_s^t) - \eta^{t-1} [\widehat{Q}_s^t - \widehat{Q}_s^{t-1}]) &\leq \ell(\pi_s, \nabla h(\pi_s^t) - \eta^{t-1} [\widehat{Q}_s^t - \widehat{Q}_s^{t-1}]) \\
-\langle \widehat{Q}_s^t - \widehat{Q}_s^{t-1}, \tilde{\pi}_s^t \rangle + {}^1\!/_{\eta^{t-1}} D(\tilde{\pi}_s^t, \pi_s^t)) &\leq -\langle \widehat{Q}_s^t - \widehat{Q}_s^{t-1}, \pi_s \rangle + {}^1\!/_{\eta^{t-1}} \left(D(\pi_s, \pi_s^t) - D_h(\pi_s, \tilde{\pi}_s^t) \right) \\
{}^1\!/_{\eta^{t-1}} D_h(\pi_s, \tilde{\pi}_s^t) &\leq -\langle \widehat{Q}_s^t - \widehat{Q}_s^{t-1}, \pi_s - \tilde{\pi}_s^t \rangle + {}^1\!/_{\eta^{t-1}} \left(D(\pi_s, \pi_s^t) - D(\tilde{\pi}_s^t, \pi_s^t) \right) \\
\implies {}^1\!/_{\eta} D_h(\pi_s, \tilde{\pi}_s^t) &\leq -{}^{\eta^{t-1}}\!/_{\eta} \langle \widehat{Q}_s^t - \widehat{Q}_s^{t-1}, \pi_s - \tilde{\pi}_s^t \rangle + {}^1\!/_{\eta} \left(D(\pi_s, \pi_s^t) - D(\tilde{\pi}_s^t, \pi_s^t) \right)
\end{aligned} \tag{19}
$$

Plugging in Eq. 18 and Eq. 19 in the definition of Lazy PMD(+correction), we can observe the objectives of PMD(+extragradient) and PMD(+correction) are related in the following way

$$
\begin{aligned}
&\ell(\pi_s, \nabla h(\tilde{\pi}_s^t) - \eta \widehat{Q}_s) - \ell(\tilde{\pi}_s^t, \nabla h(\tilde{\pi}_s^t) - \eta \widehat{Q}_s) = -\langle \widehat{Q}_s^t, \pi_s - \tilde{\pi}_s^t \rangle + {}^1\!/_{\eta} D_h(\pi_s, \tilde{\pi}_s^t) \\
&\overset{Eq.\ 18}{=} -\langle \widehat{Q}_s^t + {}^1\!/_{\eta} \nabla D_h(\tilde{\pi}_s^t, \pi_s^t), \pi_s - \tilde{\pi}_s^t \rangle + {}^1\!/_{\eta} (D(\pi_s, \pi_s^t) - D(\tilde{\pi}_s^t, \pi_s^t))
\end{aligned}
$$

$$
\begin{aligned}
&= -\langle \widehat{Q}_s^t + {}^1\!/\!_\eta \nabla D_h(\tilde{\pi}_s^t, \pi_s^t), \pi_s \rangle + {}^1\!/\!_\eta D(\pi_s, \pi_s^t) - \Big(-\langle \widehat{Q}_s^t + {}^1\!/\!_\eta \nabla D_h(\tilde{\pi}_s^t, \pi_s^t), \tilde{\pi}_s^t \rangle + {}^1\!/\!_\eta D(\tilde{\pi}_s^t, \pi_s^t) \Big) \\
&= \ell(\pi_s, \nabla h(\pi_s^t) - \eta[\widehat{Q}_s^t + {}^1\!/\!_\eta \nabla D_h(\tilde{\pi}_s^t, \pi_s^t)]) - \ell(\tilde{\pi}_s^t, \nabla h(\pi_s^t) - \eta[\widehat{Q}_s^t + {}^1\!/\!_\eta \nabla D_h(\tilde{\pi}_s^t, \pi_s^t)]) \\
&\overset{Eq.\ 19}{\leq} -\langle \widehat{Q}_s^t + {}^{\eta^{t-1}}\!/\!_\eta (\widehat{Q}_s^t - \widehat{Q}_s^{t-1}), \pi_s - \tilde{\pi}_s^t \rangle + {}^1\!/\!_\eta (D(\pi_s, \pi_s^t) - D(\tilde{\pi}_s^t, \pi_s^t)) \\
&= -\langle \widehat{Q}_s^t + {}^{\eta^{t-1}}\!/\!_\eta (Q_s^t - Q_s^{t-1}), \pi_s \rangle + {}^1\!/\!_\eta D(\pi_s, \pi_s^t) - \big(-\langle Q_s + {}^{\eta^{t-1}}\!/\!_\eta (Q_s^t - Q_s^{t-1}), \tilde{\pi}_s^t \rangle + {}^1\!/\!_\eta D(\tilde{\pi}_s^t, \pi_s^t) \big) \\
&= \ell(\pi_s, \nabla h(\pi_s^t) - \eta[\widehat{Q}_s^t + {}^{\eta^{t-1}}\!/\!_\eta (\widehat{Q}_s^t - \widehat{Q}_s^{t-1})]) - \ell(\tilde{\pi}_s^t, \nabla h(\pi_s^t) - \eta[\widehat{Q}_s^t + {}^{\eta^{t-1}}\!/\!_\eta (\widehat{Q}_s^t - \widehat{Q}_s^{t-1})])
\end{aligned}
$$

Ignoring constant terms, the ordering over objectives implies the solutions of `Lazy PMD(+momentum)` subsume those of `Lazy PMD(+correction)`. □

Lemma 12. ***(Descent Property of exact*** `PMD(+lookahead)`***)*** *Consider the policies produced by the iterative updates of* `PMD(+lookahead)`

$$\pi_s^{t+1} = \text{argmin}_\pi \langle \tilde{Q}_s^t, \pi^s \rangle + {}^1\!/\!_{\tilde{\eta}^t} D_h(\pi_s, \pi_s^t)$$

where $\tilde{Q}_s^t \doteq \mathbb{E}[r_s + \gamma \langle Q_{s'}^t, \tilde{\pi}_{s'}^t \rangle]$*,* $\tilde{\pi}_s^t$ *is greedy with respect to* Q_s^t*,* $\tilde{\eta}^t \geq 0$*. Then, for any* $t \geq 0$

$$\langle \tilde{Q}_s^t, \pi_s^{t+1} - \tilde{\pi}_s^t \rangle \geq 0, \forall s \in \mathcal{S} \tag{20}$$

$$\langle Q_s^{t+1} - \tilde{Q}_s^t, \pi_s^{t+1} \rangle \geq 0, \forall s \in \mathcal{S} \tag{21}$$

Proof. Consider first the descent property of $\tilde{\pi}^t$

$$
\begin{aligned}
\langle Q_s^t - \tilde{Q}_s^t, \tilde{\pi}_s^t \rangle &= \sum_{a \in \mathcal{A}} (Q_{s,a}^t - \tilde{Q}_{s,a}^t) \tilde{\pi}_{a|s}^t \\
&= \gamma \sum_{s' \in \mathcal{S}} \sum_{a \in \mathcal{A}} P_{s'|s,a} \tilde{\pi}_{a|s}^t \left[\langle Q_{s'}^t, \pi_{s'}^t \rangle - \langle Q_{s'}^t, \tilde{\pi}_{s'}^t \rangle \right] \leq 0
\end{aligned} \tag{22}
$$

where the last inequality follows from the definition of $\tilde{\pi}_s^t$ as greedy with respect to Q_s^t, which implies $\langle Q_{s'}^t, \tilde{\pi}_{s'}^t \rangle \geq \langle Q_{s'}^t, \pi_{s'}^t \rangle$.

Then, for the descent property of π_s^{t+1}, we have

$$
\begin{aligned}
\langle Q_s^{t+1} - \tilde{Q}_s^t, \pi_s^{t+1} \rangle &= \sum_{a \in \mathcal{A}} (Q_{s,a}^{t+1} - \tilde{Q}_{s,a}^t) \pi_{a|s}^{t+1} \\
&= \gamma \sum_{s' \in \mathcal{S}} \sum_{a \in \mathcal{A}} P_{s'|s,a} \pi_{a|s}^{t+1} \left[\langle Q_{s'}^{t+1}, \pi_{s'}^{t+1} \rangle - \langle Q_{s'}^t, \tilde{\pi}_{s'}^t \rangle \right] \\
&= \gamma \sum_{s' \in \mathcal{S}} \sum_{a \in \mathcal{A}} P_{s'|s,a} \pi_{a|s}^{t+1} \left[\langle Q_{s'}^{t+1}, \pi_{s'}^{t+1} \rangle - \langle \tilde{Q}_{s'}^t, \tilde{\pi}_{s'}^t \rangle - \langle Q_{s'}^t - \tilde{Q}_{s'}^t, \tilde{\pi}_{s'}^t \rangle \right] \\
&\overset{(Eq.\ 22)}{\geq} \gamma \sum_{s' \in \mathcal{S}} \sum_{a \in \mathcal{A}} P_{s'|s,a} \pi_{a|s}^{t+1} \left[\langle Q_{s'}^{t+1}, \pi_{s'}^{t+1} \rangle - \langle \tilde{Q}_{s'}^t, \tilde{\pi}_{s'}^t \rangle \right] \\
&= \gamma \sum_{s'} \sum_{a} P_{s'|s,a} \pi_{a|s}^{t+1} \left[\langle \tilde{Q}_{s'}^t, \pi_{s'}^{t+1} - \tilde{\pi}_{s'}^t \rangle + \langle Q_{s'}^{t+1} - \tilde{Q}_{s'}^t, \pi_{s'}^{t+1} \rangle \right]
\end{aligned}
$$

Recursing, yields

$$\langle Q_s^{t+1} - \tilde{Q}_s^t, \pi_s^{t+1} \rangle = {}^\gamma\!/\!_{1-\gamma} \sum_{s'} d_{s' \to s'}^{t+1} \langle \tilde{Q}_{s'}^t, \pi_{s'}^{t+1} - \tilde{\pi}_{s'}^t \rangle \tag{23}$$

From Lemma 4 with $\pi_s = \tilde{\pi}_s^t$ and $\tilde{\eta}^t \geq 0$

$$\langle \tilde{Q}_s^t, \pi_s^{t+1} - \tilde{\pi}_s^t \rangle \geq {}^1\!/\!_{\tilde{\eta}^t} \left(D_h(\tilde{\pi}_s^t, \pi_s^{t+1}) + D_h(\pi_s^{t+1}, \tilde{\pi}_s^t) \right) \geq 0 \tag{24}$$

This proves the first claim in Eq. 20. Plugging Eq. 24 back in Eq. 23 we show the second claim in Eq. 21, $\langle Q_s^{t+1} - \tilde{Q}_s^t, \pi_s^{t+1} \rangle \geq 0$ □

Proposition 13. ***(Functional acceleration with*** `PMD(+lookahead)`***)*** *Consider the policies produced by the iterative updates of* `PMD(+lookahead)`

$$\pi_s^{t+1} = \text{argmin}_{\pi_s \in \Delta(\mathcal{A})} -\langle \tilde{Q}_s^t, \pi_s \rangle + {}^1\!/\!_{\tilde{\eta}^t} D_h(\pi_s, \pi_s^t) \tag{25}$$

where $\tilde{Q}_s^t \doteq \mathbb{E}[r_s + \gamma\langle\widehat{Q}_{s'}^t, \tilde{\pi}_{s'}^t\rangle]$, $\tilde{\pi}_s^t$ *is greedy with respect to* Q_s^t, $\tilde{\eta}^t \geq 0$ *are adaptive step sizes, such that* $\forall\epsilon_t$ *arbitrarily small,* $\tilde{\eta}^t \geq 1/\epsilon_t D_h(\mathrm{greedy}(\tilde{Q}_s^t), \pi_s^t)$. *Then,*

$$V_s^* - V_s^{t+1} \leq \gamma^2\|V^* - V^t\|_\infty + \epsilon_t \tag{26}$$

and recursing yields

$$\|V^* - V^t\|_\infty \leq (\gamma^2)^t(\|V^* - V^0\|_\infty + \sum_{i\leq t} \epsilon_i/(\gamma^2)^i) \tag{27}$$

Proof. If π_{t+1} is the result of a PMD update which uses $\tilde{Q}_s^t$, and step-sizes $\tilde{\eta}^t$, then applying Lemma 4 for $\pi_s = \bar{\pi}_s^{t+1}$ greedy with respect to $\tilde{Q}_s^t$

$$-\langle\tilde{Q}_s^t, \pi_s^{t+1} - \bar{\pi}_s^{t+1}\rangle \leq 1/\tilde{\eta}^t \left(D_h(\bar{\pi}_s^{t+1}, \pi_s^t) - D_h(\pi_s^{t+1}, \pi_s^t) - D_h(\pi_s^{t+1}, \bar{\pi}_s^{t+1})\right) \tag{28}$$

$$\leq 1/\tilde{\eta}^t D_h(\bar{\pi}_s^{t+1}, \pi_s^t) \tag{29}$$

Further, the suboptimality is

$$V_s^* - V_s^{t+1} = -\langle\tilde{Q}_s^t, \pi_s^{t+1} - \pi_s^*\rangle - \langle Q_s^{t+1} - \tilde{Q}_s^t, \pi_s^{t+1}\rangle + \langle Q_s^* - \tilde{Q}_s^t, \pi_s^*\rangle \tag{30}$$

Since $\langle\tilde{Q}_s^t, \pi_s^*\rangle \leq \langle\tilde{Q}_s^t, \bar{\pi}_s^{t+1}\rangle$ if $\bar{\pi}_s^{t+1}$ is greedy with respect to $\tilde{Q}_s^t$, then plugging Eq 29 in Eq 30

$$V_s^* - V_s^{t+1} \leq 1/\eta^t D_h(\bar{\pi}_s^{t+1}, \pi_s^t) - \langle Q_s^{t+1} - \tilde{Q}_s^t, \pi_s^{t+1}\rangle + \langle Q_s^* - \tilde{Q}_s^t, \pi_s^*\rangle \tag{31}$$

Next, cf. Lemma 12, $\langle Q_s^{t+1} - \tilde{Q}_s^t, \pi_s^{t+1}\rangle \geq 0$. Plugging this back into Eq 30

$$\begin{aligned} V_s^* - V_s^{t+1} &\leq 1/\tilde{\eta}^t D_h(\bar{\pi}_s^{t+1}, \pi_s^t) + \langle Q_s^* - \tilde{Q}_s^t, \pi_s^*\rangle \\ &\leq \langle Q_s^* - \tilde{Q}_s^t, \pi_s^*\rangle + \epsilon_t \end{aligned} \tag{32}$$

where the last step follows from step-size adaptation condition $\tilde{\eta}^t \geq 1/\epsilon_t D_h(\bar{\pi}_s^{t+1}, \pi_s^t)$.

Decomposing the remaining term

$$\begin{aligned} \langle Q_s^* - \tilde{Q}_s^t, \pi_s^*\rangle &= \sum_{a\in\mathcal{A}}(Q_{s,a}^* - \tilde{Q}_{s,a}^t)\pi_{a|s}^* \\ &= \gamma\sum_{s'\in\mathcal{S}}\sum_{a\in\mathcal{A}} P_{s'|s,a}\pi_{a|s}^*\left[\langle Q_{s'}^*, \pi_{s'}^*\rangle - \langle Q_{s'}^t, \tilde{\pi}_{s'}^t\rangle\right] \\ &= \gamma\sum_{s'\in\mathcal{S}}\sum_{a\in\mathcal{A}} P_{s'|s,a}\pi_{a|s}^*\left[\langle Q_{s'}^* - Q_{s'}^t, \pi_{s'}^*\rangle - \langle Q_{s'}^t, \tilde{\pi}_{s'}^t - \pi_{s'}^*\rangle\right] \\ &\leq \gamma\sum_{s'\in\mathcal{S}}\sum_{a\in\mathcal{A}} P_{s'|s,a}\pi_{a|s}^*\left[\langle Q_{s'}^* - Q_{s'}^t, \pi_{s'}^*\rangle\right] \\ &= \gamma^2\sum_{s'\in\mathcal{S}}\sum_{a\in\mathcal{A}}\pi_{a|s}^* P_{s'|s,a}\sum_{s''\in\mathcal{S}}\sum_{a'\in\mathcal{A}}\pi_{s',a'}^* P_{s''|s',a'}\left[V_{s''}^* - V_{s''}^t\right] \end{aligned}$$

where the inequality follows due to $\tilde{\pi}_s^t$ being greedy with respect to Q_s^t, $\forall s \in \mathcal{S}$ by definition.

Taking the max norm and applying the triangle inequality and the contraction property

$$\|\langle Q^* - \tilde{Q}^t, \pi^*\rangle\|_\infty \leq \gamma^2\|V^* - V^t\|_\infty$$

and then plugging this back in Eq. 32

$$\begin{aligned} V_s^* - V_s^{t+1} &\leq \|\langle Q_s^* - \tilde{Q}^t, \pi^*\rangle\|_\infty + \epsilon_t \\ &\leq \gamma^2\|V^* - V^t\|_\infty + \epsilon_t \end{aligned}$$

which is the first claim in Eq.26. Then recursing yields the second claim in Eq.27. □

Proposition 14. ***(Functional acceleration with*** `PMD(+extragradient)`***)*** *Consider the policies produced by the iterative updates of* `PMD(+extragradient)`

$$\tilde{\pi}_s^t = \mathrm{argmin}_{\pi_s\in\Delta(\mathcal{A})} -\langle\widehat{Q}_s^t, \pi_s\rangle + 1/\eta^t D_h(\pi_s, \pi_s^t), \quad \tilde{Q}_s^t \doteq \mathbb{E}[r_s + \gamma\langle\widehat{Q}_{s'}^t, \tilde{\pi}_{s'}^t\rangle] \tag{33}$$

$$\pi_s^{t+1} = \text{argmin}_{\pi_s \in \Delta(\mathcal{A})} -\langle \tilde{Q}_s^t, \pi_s \rangle + {}^1\!/\!_{\tilde{\eta}^t} D_h(\pi_s, \pi_s^t) \tag{34}$$

where $\tilde{\eta}^t, \eta^t \geq 0$ *are adaptive step sizes, such that* $\forall \epsilon_t, \tilde{\epsilon}_t$ *arbitrarily small,* $\tilde{\eta}^t \geq {}^1\!/\!_{\epsilon_t} D_h(\text{greedy}(\tilde{Q}_s^t), \pi_s^t)$ *and* $\eta^t \geq {}^1\!/\!_{\bar{\epsilon}_t} D_h(\text{greedy}(Q^t), \pi_s^t)$. *Then*

$$V_s^* - V_s^{t+1} \leq \gamma^2 \|V^* - V^t\|_\infty + \gamma \tilde{\epsilon}_t + \epsilon_t \tag{35}$$

and recursing yields

$$\|V_s^* - V_s^t\|_\infty \leq (\gamma^2)^t (\|V^* - V^0\|_\infty + \sum_{i \leq t} {}^{(\epsilon_i + \gamma \tilde{\epsilon}_t)}\!/\!_{(\gamma^2)^i}) \tag{36}$$

Proof. If π_{t+1} is the result of a PMD update with $\tilde{Q}_s^t$, applying Lemma 4 for $\bar{\pi}^{t+1}$ greedy with respect to $\tilde{Q}_s^t$

$$-\langle \tilde{Q}_s^t, \pi_s^{t+1} - \bar{\pi}_s^{t+1} \rangle \leq {}^1\!/\!_{\tilde{\eta}^t} \left(D_h(\bar{\pi}_s^{t+1}, \pi_s^t) - D_h(\pi_s^{t+1}, \pi_s^t) - D_h(\pi_s^{t+1}, \bar{\pi}_s^{t+1}) \right) \tag{37}$$

$$\leq {}^1\!/\!_{\tilde{\eta}^t} D_h(\bar{\pi}_s^{t+1}, \pi_s^t) \tag{38}$$

Further, the suboptimality is

$$V_s^* - V_s^{t+1} = -\langle \tilde{Q}_s^t, \pi_s^{t+1} - \pi_s^* \rangle - \langle Q_s^{t+1} - \tilde{Q}_s^t, \pi_s^{t+1} \rangle + \langle Q_s^* - \tilde{Q}_s^t, \pi_s^* \rangle \tag{39}$$

Since $\langle \tilde{Q}_s^t, \pi_s^* \rangle \leq \langle \tilde{Q}_s^t, \bar{\pi}_s^{t+1} \rangle$ if $\bar{\pi}_s^{t+1}$ is greedy with respect to $\tilde{Q}_s^t$, then plugging Eq 38 in Eq 39 we have

$$V_s^* - V_s^{t+1} \leq {}^1\!/\!_{\eta^t} D_h(\bar{\pi}_s^{t+1}, \pi_s^t) - \langle Q_s^{t+1} - \tilde{Q}_s^t, \pi_s^{t+1} \rangle + \langle Q_s^* - \tilde{Q}_s^t, \pi_s^* \rangle \tag{40}$$

Next, cf. Lemma 12, $\langle Q_s^{t+1} - \tilde{Q}_s^t, \pi_s^{t+1} \rangle \geq 0$ Plugging back into Eq 39, we obtain

$$\begin{aligned} V_s^* - V_s^{t+1} &\leq {}^1\!/\!_{\eta^t} D_h(\bar{\pi}_s^{t+1}, \pi_s^t) + \langle Q_s^* - \tilde{Q}_s^t, \pi_s^* \rangle \\ &\leq \langle Q_s^* - \tilde{Q}_s^t, \pi_s^* \rangle + \epsilon_t \end{aligned} \tag{41}$$

where the last step follows from step-size adaptation condition.

Applying Lemma 4 for $\check{\pi}_s^{t+1}$ greedy with respect to Q^t,

$$\begin{aligned} -\langle Q_s^t, \tilde{\pi}_s^t - \pi_s^* \rangle \leq -\langle Q_s^t, \tilde{\pi}_s^t - \check{\pi}_s^{t+1} \rangle &\leq {}^1\!/\!_{\eta^t} \left(D_h(\check{\pi}_s^{t+1}, \pi_s^t) - D_h(\pi_s^{t+1}, \pi_s^t) - D_h(\pi_s^{t+1}, \check{\pi}_s^{t+1}) \right) \\ &\leq {}^1\!/\!_{\eta^t} D_h(\check{\pi}_s^{t+1}, \pi_s^t) \end{aligned} \tag{42}$$

Further,

$$\begin{aligned} \langle Q_s^* - \tilde{Q}_s^t, \pi_s^* \rangle &= \sum_{a \in \mathcal{A}} (Q_{s,a}^* - \tilde{Q}_{s,a}^t) \pi_{a|s}^* \\ &= \gamma \sum_{s' \in \mathcal{S}} \sum_{a \in \mathcal{A}} P_{s'|s,a} \pi_{a|s}^* \left[\langle Q_{s'}^*, \pi_{s'}^* \rangle - \langle Q_{s'}^t, \tilde{\pi}_{s'}^t \rangle \right] \\ &= \gamma \sum_{s' \in \mathcal{S}} \sum_{a \in \mathcal{A}} P_{s'|s,a} \pi_{a|s}^* \left[\langle Q_{s'}^* - Q_{s'}^t, \pi_{s'}^* \rangle - \langle Q_{s'}^t, \tilde{\pi}_{s'}^t - \pi_{s'}^* \rangle \right] \\ &\overset{Eq.\ 42}{\leq} \gamma \sum_{s' \in \mathcal{S}} \sum_{a \in \mathcal{A}} P_{s'|s,a} \pi_{a|s}^* \left[\langle Q_{s'}^* - Q_{s'}^t, \pi_{s'}^* \rangle + {}^1\!/\!_{\eta^t} D_h(\check{\pi}_s^{t+1}, \pi_s^t) \right] \\ &\overset{\text{cf. premise}}{\leq} \gamma \sum_{s' \in \mathcal{S}} \sum_{a \in \mathcal{A}} P_{s'|s,a} \pi_{a|s}^* \left[\langle Q_{s'}^* - Q_{s'}^t, \pi_{s'}^* \rangle + \tilde{\epsilon}^t \right] \\ &= \gamma^2 \sum_{s' \in \mathcal{S}} \sum_{a \in \mathcal{A}} \pi_{a|s}^* P_{s'|s,a} \left[\sum_{s'' \in \mathcal{S}} \sum_{a' \in \mathcal{A}} \pi_{s',a'}^* P_{s''|s',a'} \left[V_{s''}^* - V_{s''}^t \right] + \tilde{\epsilon}^t \right] \end{aligned}$$

Taking the max norm, using the triangle inequality and contraction property

$$\|\langle Q^* - \tilde{Q}^t, \pi^* \rangle\|_\infty \leq \gamma^2 \|V^* - V^t\|_\infty + \gamma \tilde{\epsilon}_t$$

Plugging back in Eq. 41

$$\begin{aligned} V_s^* - V_s^{t+1} &\leq \langle Q_s^* - \tilde{Q}_s^t, \pi_s^* \rangle + \epsilon_t \\ &\leq \|\langle Q^* - \tilde{Q}^t, \pi^* \rangle\|_\infty + \epsilon_t \\ &\leq \gamma^2 \|V^* - V^t\|_\infty + \gamma \tilde{\epsilon}_t + \epsilon_t \end{aligned}$$

which is the first claim in Eq.35. Recursing yields the second claim in Eq.36. □

C Details on PMD updates for Sec. 4: Functional Acceleration for PMD

Algorithm 2 PMD(++)

1: Input: $\tilde{\pi}_0, \pi_0 \in \operatorname{rint}\Pi$, adaptive $\{\tilde{\eta}^t, \eta^t\}_{t\geq 0}$
2: **for** $t = 1, 2 \dots T$ **do**
3: PMD(+lookahead)

$$\tilde{\pi}_s^t = \operatorname{argmin}_{\pi_s \in \Delta(\mathcal{A})} -\langle Q_s^t, \pi_s\rangle, \quad \tilde{Q}_s^t \doteq \mathbb{E}[r_s + \gamma\langle Q_{s'}^t, \tilde{\pi}_{s'}^t\rangle]$$
$$\pi_s^{t+1} = \operatorname{argmin}_{\pi_s \in \Delta(\mathcal{A})} -\langle \tilde{Q}_s^t, \pi_s\rangle + {}^1\!/_{\tilde{\eta}^t} D_h(\pi_s, \pi_s^t)$$

4: PMD(+extragradient)

$$\tilde{\pi}_s^t = \operatorname{argmin}_{\pi_s \in \Delta(\mathcal{A})} -\langle Q_s^t, \pi_s\rangle + {}^1\!/_{\eta^t} D_h(\pi_s, \pi_s^t), \quad \tilde{Q}_s^t \doteq \mathbb{E}[r_s + \gamma\langle Q_{s'}^t, \tilde{\pi}_{s'}^t\rangle]$$
$$\pi_s^{t+1} = \operatorname{argmin}_{\pi_s \in \Delta(\mathcal{A})} -\langle \tilde{Q}_s^t, \pi_s\rangle + {}^1\!/_{\tilde{\eta}^t} D_h(\pi_s, \pi_s^t)$$

5: PMD(+correction)

$$\tilde{\pi}_s^t = \operatorname{argmin}_{\pi_s \in \Delta(\mathcal{A})} -\langle Q_s^t, \pi_s\rangle + {}^1\!/_{\eta^t} D_h(\pi_s, \pi_s^t), \quad \tilde{Q}_s^t \doteq \mathbb{E}[r_s + \gamma\langle Q_{s'}^t, \tilde{\pi}_{s'}^t\rangle]$$
$$\pi_s^{t+1} = \operatorname{argmin}_{\pi_s \in \Delta(\mathcal{A})} -\langle \tilde{Q}_s^t - {}^{\eta_t}\!/_{\tilde{\eta}_t} Q_s^t, \pi_s\rangle + {}^1\!/_{\tilde{\eta}^t} D_h(\pi_s, \tilde{\pi}_s^t)$$

6: Lazy PMD(+correction)

$$\tilde{\pi}_s^t = \operatorname{argmin}_{\pi_s \in \Delta(\mathcal{A})} -\langle Q_s^t - Q_s^{t-1}, \pi_s\rangle + {}^1\!/_{\eta_{t-1}} D_h\left(\pi_s, \pi_s^t\right)$$
$$\pi_s^{t+1} = \operatorname{argmin}^{\pi_s \in \Delta(\mathcal{A})} \langle Q_s^t, \pi_s\rangle + {}^1\!/_{\tilde{\eta}^t} D_h(\pi_s, \tilde{\pi}_s^t)$$

7: Lazy PMD(+momentum)

$$\pi_s^{t+1} = \operatorname*{argmin}_{\pi_s \in \Delta(\mathcal{A})} \langle Q_s^t + {}^{\eta_{t-1}}\!/_{\eta_t}(Q_s^t - Q_s^{t-1}), \pi_s\rangle + {}^1\!/_{\eta^t} D_h(\pi_s, \pi_s^t)$$

8: **end for**

D Details on Algorithmic Implementation for Sec. 4.1: Approximate Functional Acceleration for Parametric Policies

We use the following shorthand notation for the updates PMD, PMD(+ext), PMD(+cor), PMD(+lzc), PMD(+mom).

Policy approximation We parametrize the policy iterates using a Bregman policy class $\{\pi^\theta : \pi_s^\theta = \operatorname{proj}^h_{\Delta(\mathcal{A})}(\nabla h^*(f_s^\theta)), s \in \mathcal{S}\}$ with a tabular parametrization θ. For the updates requiring two policies, we keep them parametrized separately with $\tilde{\pi}_{\mathbf{w}}$ and π_θ.

We formulate the policy optimization problem using the extension proposed by Tomar et al. (2020). Each iteration, in an "inner-loop" optimization procedure, we update θ and $\mathbf{w}$ using k and n, respectively, updates with standard GD on the composite PMD surrogate model, denoted $\ell : \Theta \to \mathbb{R}$ (with $\Theta \doteq \mathbb{R}^{|\mathcal{S}|\times|\mathcal{A}|}$ cf. the tabular parametrization) associated with the policy represented by those parameters π_θ, or $\tilde{\pi}_{\mathbf{w}}$, respectively. We execute the parameter optimization in expectation over the state-action space. Concretely, for PMD we use the surrogate

$$\ell(\theta) \doteq \mathbb{E}_{s\sim d_\rho^t}\left[-\langle \widehat{Q}_s^t, \pi^\theta\rangle + {}^1\!/_{\eta_s^t} D_h(\pi_s^\theta, \pi_s^t)\right] \tag{43}$$

and update in an "inner-loop" optimization procedure

(init) $\theta^{(0)} \doteq \theta_t$ *(for* $i \in [0..k]$*)* $\theta^{(i+1)} = \theta^{(i)} - \beta\nabla_{\theta^{(i)}}\ell(\theta^{(i)})$ *(final)* $\theta_{t+1} \doteq \theta^{(k)}$

with β—a small learning rate. The optimization procedure for $\mathbf{w}$ is analogous. The rest of the algorithms use the surrogate objectives as described in Sec. 4.

We use state dependent step-sizes, and for step-size adaptation, we compute $\eta_s^t = {}^{D_h(\text{greedy}(\widehat{Q}_s^t),\pi_s^t)}/_{\gamma^t\epsilon_0}$, with $\epsilon_0 = 10^{-4}$ a small constant according to the optimal adaptive schedule for PMD, cf. Johnson et al. (2023). The value for $\tilde{\eta}_s^t$ is chosen analogously. Instead of using $\tilde{Q}_s^t \doteq \mathbb{E}[r_s + \gamma\langle\widehat{Q}_{s'}^t, \tilde{\pi}_t\rangle]$ Sec. 4, in the numerical studies, we used $\tilde{Q}_s^t \doteq Q_s^{\tilde{\pi}_t}$ or approximations thereof in the inexact settings. Similarly, $\widehat{Q}_s^t \doteq Q_s^{\pi_t}$ or approximations thereof.

Objectives We now describe in detail the objectives for each algorithm.

PMD—We optimize approximately the objective in Eq.43 with respect to the parameters θ of π^θ with k GD updates using a per-state step-size $\eta_s^t = {}^{D_h(\text{greedy}(\widehat{Q}_s^t),\pi_s^t)}/_{\gamma^t\epsilon_0}$.

PMD(+loo)—We keep two policies $\tilde{\pi}$, π^θ—the former is the non-parametric greedy policy

$$\tilde{\pi}_s^t = \max_{a\in\mathcal{A}} \widehat{Q}_s^t(a)$$

The latter is parametrized and its parameters θ optimize the following lookahead-based surrogate with n GD updates using step-size adaptation $\tilde{\eta}_s^t = {}^{D_h(\text{greedy}(\tilde{Q}_s^t),\pi_s^t)}/_{\gamma^t\epsilon_0}$.

$$\ell(\theta) \doteq \mathbb{E}_{s\sim d_\rho^t}\left[-\langle\tilde{Q}_s^t, \pi_s^\theta\rangle + {}^1/_{\tilde{\eta}_s^t} D_h(\pi_s^\theta, \pi_s^t)\right]$$

PMD(+ext)—The update to π^θ is identical to PMD(+loo). In contrast to PMD(+lookahead), $\tilde{\pi}^{\mathbf{w}}$ is parametrized with parameter vector $\mathbf{w}$. The update to $\tilde{\pi}_s^{\mathbf{w}}$ uses, at each iteration, k GD updates on the surrogate objective

$$\ell(\mathbf{w}) \doteq \mathbb{E}_{s\sim d_\rho^t}\left[-\langle\widehat{Q}_s^t, \tilde{\pi}_s^{\mathbf{w}}\rangle + {}^1/_{\eta_s^t} D_h(\tilde{\pi}_s^{\mathbf{w}}, \pi_s^t)\right]$$

The step-sizes η_s^t are adapted using $\eta_s^t = {}^{D_h(\text{greedy}(\widehat{Q}_s^t),\pi_s^t)}/_{\gamma^t\epsilon_0}$

PMD(+cor)—Identical to PMD(+ext) in all aspects except the update to the parameter vector θ of π_s^θ, which is now updated, at each iteration, using n GD updates on the objective

$$\ell(\theta) \doteq \mathbb{E}_{s\sim d_\rho^t}\left[-\langle[\tilde{Q}_s^t - {}^{\eta_s^t}/_{\tilde{\eta}_s^t} Q_s^t], \pi_s^\theta\rangle + {}^1/_{\tilde{\eta}_s^t} D_h(\pi_s^\theta, \tilde{\pi}_s^t)\right]$$

where $\tilde{\pi}_s^t \doteq \tilde{\pi}_s^{\mathbf{w}_t}$.

PMD(+lzc)—Each iteration, the parameter vector $\mathbf{w}$ is updated using n GD updates on the objective

$$\ell(\mathbf{w}) \doteq \mathbb{E}_{s\sim d_\rho^t}\left[-\langle\widehat{Q}_s^t - \widehat{Q}_s^{t-1}, \tilde{\pi}_s^{\mathbf{w}}\rangle + {}^1/_{\eta_s^{t-1}} D_h(\tilde{\pi}_s^{\mathbf{w}}, \pi_s^t)\right]$$

Each iteration, the parameter vector θ is updated using k GD updates on the objective

$$\ell(\theta) \doteq \mathbb{E}_{s\sim d_\rho^t}\left[-\langle\widehat{Q}_s^t, \pi_s^\theta\rangle + {}^1/_{\eta_s^t} D_h(\pi_s^\theta, \tilde{\pi}_s^t)\right]$$

PMD(+mom)—A single set of parameters θ are learned by updating, at each iteration, using $(k+n)$ GD updates on the objective

$$\ell(\theta) \doteq \mathbb{E}_{s\sim d_\rho^t}\left[-\langle\widehat{Q}_s^t + {}^{\eta_s^{t-1}}/_{\eta_s^t}[\widehat{Q}_s^t - \widehat{Q}_s^{t-1}], \pi_s^\theta\rangle + {}^1/_{\eta_s^t} D_h(\pi_s^\theta, \pi_s^t)\right]$$

E Approximate Policy Mirror Descent as Projected Gradient Descent (PGD)

In this section we provide an alternative perspective on PMD—cf. Lemma 3, stating that the MD update can be rewritten in the following ways

$$\begin{aligned} x_{t+1} &= \text{argmin}_{x\in\mathcal{X}\cap\mathcal{C}}\, D_h(x, \nabla h^*(\nabla h(x_t) + \eta\nabla f(x_t))) && \text{(PGD)} \\ &= \text{argmin}_{x\in\mathcal{X}\cap\mathcal{C}}\, \eta\,\langle\nabla f(x_t), x\rangle + D_h(x, x_t) && \text{(proximal perspective)} \end{aligned}$$

Algorithm 3 `Approximate Lazy PMD(+momentum)` (PGD perspective)

1: Initialize policy parameter $\theta_0 \in \Theta$, mirror map h, small constant ϵ_0, learning rate β
2: **for** $t = 1, 2 \dots T$ **do**
3: Find $\widehat{Q}^t$ approximating Q^t (critic update)
4: Compute adaptive step-size $\eta^t = {}^{D_h(\text{greedy}(\widehat{Q}^t)}/_{\gamma^{2(t+1)}\epsilon_0}$
5: Find $\pi^{t+1} \doteq \pi^{\theta_{t+1}} = \nabla h^*(f^{\theta_{t+1}})$ by (approximately) solving the surrogate problem (with k GD updates)
6: $\min_{\theta\in\Theta} \ell(\theta)$ $\quad \ell(\theta) \doteq -\mathbb{E}_{s\sim d^t_\rho}[D_h(\pi^\theta_s, \nabla h^*(\nabla h(\pi^t_s) - \eta^t \widehat{Q}^t_s - \eta^{t-1}(\widehat{Q}^t_s - \widehat{Q}^{t-1}_s))]$
7: *(init)* $\theta^{(0)} \doteq \theta_t$ $\quad$ *(for* $i \in [0..k-1]$*)* $\theta^{(i+1)} = \theta^{(i)} - \beta\nabla_{\theta^{(i)}}\ell(\theta^{(i)})$ $\quad$ *(final)* $\theta_{t+1} \doteq \theta^{(k)}$
8: **end for**

Alg. 3 describes a PGD perspective on `Lazy PMD(+momentum)`, following Alfano et al. (2024); Haarnoja et al. (2018); Abdolmaleki et al. (2018). At each iteration, after taking a gradient step in dual space, a Bregman projection is used on the dual approximation mapped back to the policy space, to satisfy the simplex constraint.

F Newton's method

The Newton-Kantorovich theorem generalizes Newton's method for solving nonlinear equations to infinite-dimensional Banach spaces. It provides conditions under which Newton's method converges and gives an estimate of the convergence rate. Newton-Kantorovich theorem deals with the convergence of Newton's method for a nonlinear operator $F : \mathcal{X} \to \mathcal{Y}$, where $\mathcal{X}$ and $\mathcal{Y}$ are Banach spaces. The method iteratively solves $F(x) = 0$ using

$$x^{t+1} = x^t - (\nabla F)^{-1} F(x^t)$$

where ∇F is a generalization of the Jacobian of F, provided F is differentiable. Intuitively, at each iteration, the method performs a linearization of $F(x) = 0$ close to x, using a first order Taylor expansion: $F(x + \Delta x) \approx F(x) + \nabla F(x)\Delta x$, where $F(x) + \nabla F(x)\Delta x = 0 \iff \Delta x = -(\nabla F)^{-1}F(x)$. For x close to x^*, $x - (\nabla F)^{-1}F(x)$ is a good approximation of x^*. The iterative sequence $\{x^t\}_{t\geq 0}$ converges to x^*, assuming the Jacobian matrix exists, is invertible, and Lipschitz continuous.

Quasi-Newton methods Any method that replaces the exact computation of the Jacobian matrices in the Newton's method (or their inverses) with an approximation, is a quasi-Newton method. A quasi-Newton method constructs a sequence of iterates $\{x^t\}_{t\geq 0}$ and a sequence of matrices $\{J^t\}_{t\geq 0}$ such that J^t is an approximation of the Jacobian $\nabla F(x^t)$ for any $t \geq 0$ and

$$x^{t+1} = x^t - (J^t)^{-1} F(x^t)$$

In Anderson's acceleration (Anderson, 1965), information about the last iterates is used to update the approximation of J^t.

Policy iteration as Newton's method In the context of Markov Decision Processes (MDPs), policy iteration may be interpreted as Newton's method with the following notations and analogies. First, using the Bellman optimality operator $\mathcal{T}V_s \doteq \max_a[r_{s,a} + \gamma \sum_{s'} P_{s'|s,a}V_{s'}]$, the aim is to find V such that $V = \mathcal{T}V$, which is akin to finding the roots V, such that $F(V) = V - \mathcal{T}V = (I - \mathcal{T})(V) = 0$. We interpret $F = \nabla f$ as the gradient of an unknown function $f : \mathbb{R}^n \to \mathbb{R}^n$, despite the Bellman operator being non-differentiable in general due to the max. Where the greedy policy π^t attains the max in $\mathcal{T}V$, we obtain $J^t = I - \gamma P^{\pi^t}$, which is invertible for $\gamma \in (0, 1)$. Expanding the Bellman operator, we have

$$V^{\pi^{t+1}} = r^{\pi^{t+1}} + \gamma P^{\pi^{t+1}} V^{\pi^{t+1}} \implies V^{\pi^{t+1}} = (J^t)^{-1} r^{\pi^{t+1}}$$
$$J^t V^{\pi^{t+1}} = r^{\pi^{t+1}}$$

The values corresponding to two successive PI steps can be related in the following way by manipulating the equations (Puterman and Brumelle, 1979; Grand-Clément, 2021)

$$V^{\pi^{t+1}} = (J^t)^{-1} r^{\pi^{t+1}}$$

$$
\begin{aligned}
&= V^{\pi^t} - V^{\pi^t} + (J^t)^{-1} r^{\pi^{t+1}} \\
&= V^{\pi^t} - (J^t)^{-1} J^t V^{\pi^t} + (J^t)^{-1} r^{\pi^{t+1}} \\
&= V^{\pi^t} - (J^t)^{-1} (-r^{\pi^{t+1}} + J^t V^{\pi^t}) \\
&= V^{\pi^t} - (J^t)^{-1} (-r^{\pi^{t+1}} + (I - \gamma P^{\pi^{t+1}}) V^{\pi^t}) \\
&= V^{\pi^t} - (J^t)^{-1} (V^{\pi^t} - r^{\pi^{t+1}} - \gamma P^{\pi^{t+1}} V^{\pi^t}) \\
&= V^{\pi^t} - (J^t)^{-1} (V^{\pi^t} - \mathcal{T} V^{\pi^t})
\end{aligned}
$$

In the main text we used the notation $\Psi^t \doteq (I - \gamma P^{\pi^t})^{-1} = (J^t)^{-1}$, and applied the definition $\nabla f(V^{\pi^t}) \doteq F(V^{\pi^t}) = (I - \mathcal{T})(V^{\pi^t}) = V^{\pi^t} - \mathcal{T} V^{\pi^t}$ which yielded the expression

$$
V^{\pi^{t+1}} = V^{\pi^t} - \Psi \nabla f(V^{\pi^t})
$$

G Experimental details for Sec. 5: Numerical Studies

G.1 Details of two-state Markov Decision Processes

In this section we give the specifics of the two-state MDPs presented in this work. We make use of the notation

$$P(s_k|s_i, a_j) = \mathbf{P}[i \times |\mathcal{A}| + j][k], \text{ with } \mathbf{P} \in \mathbb{R}^{|\mathcal{S}||\mathcal{A}|\times|\mathcal{S}|}$$
$$r(s_i, a_j) = \mathbf{r}[i \times |\mathcal{A}| + j], \text{ with } \mathbf{r} \in \mathbb{R}^{|\mathcal{S}||\mathcal{A}|}$$

In the main text, we used example (i), and in Appendix H.2, additionally examples (ii), (iii), and (iv)

(i) $|\mathcal{A}| = 2, \gamma = 0.9, \mathbf{r} = [-0.45, -0.1, 0.5, 0.5]$,
$\mathbf{P} = [[-0.45, 0.3], [0.99, 0.01], [0.2, 0.8], [0.99, 0.01]]$

(ii) $|\mathcal{A}| = 2, \gamma = 0.9, \mathbf{r} = [0.06, 0.38, -0.13, 0.64]$,
$\mathbf{P} = [[0.01, 0.99], [0.92, 0.08], [0.08, 0.92], [0.70, 0.30]]$

(iii) $|\mathcal{A}| = 2, \gamma = 0.9, \mathbf{r} = [0.88, -0.02, -0.98, 0.42]$,
$\mathbf{P} = [[0.96, 0.04], [0.19, 0.81], [0.43, 0.57], [0.72, 0.28]]$

(iv) $|\mathcal{A}| = 3, \gamma = 0.8, \mathbf{r} = [-0.1, -1., 0.1, 0.4, 1.5, 0.1]$,
$\mathbf{P} = [[0.9, 0.1], [0.2, 0.8], [0.7, 0.3], [0.05, 0.95], [0.25, 0.75], [0.3, 0.7]]$

G.2 Details of Random Markov Decision Processes

We consider randomly constructed finite MDPs—`Random MDP` problems (a.k.a. Garnet MDPs: Generalized Average Reward Non-stationary Environment Test-bench) (Archibald et al., 1995; Bhatnagar et al., 2009), abstract, yet representative of the kind of MDP encountered in practice, which serve as a test-bench for RL algorithms (Goyal and Grand-Clement, 2021; Scherrer and Geist, 2014; Vieillard et al., 2019). A `Random MDP` generator $\mathcal{M} \doteq (|\mathcal{S}|, |\mathcal{A}|, b, \gamma)$ is parameterized by 4 parameters: number of states $|\mathcal{S}|$, number of actions $|\mathcal{A}|$, branching factor b specifying how many possible next states are possible for each state-action pair.

The transition probabilities $P(s_0|s, a)$ are then computed as follows. First, b states $(s_1, \dots s_b)$ are chosen uniformly at random and transition probabilities are set by sampling uniform random $b-1$ numbers (cut points) between 0 and 1 and sorted as $(p_0 = 0, p_1, \dots p_{b-1}, p_b = 1)$. Then, the transition probabilities are assigned as $P(s_i|s, a) = p_i - p_{i-1}$ for each $1 \leq i \leq b$. The reward is state-dependent, and for each MDP, the per-state reward r_s is uniformly sampled between 0 and $R_{\max}$, such that $r \sim (0, R_{\max})^{|\mathcal{S}|}$. The illustrations shown use $|\mathcal{S}| = 100$ and $R_{\max} = 100$. Other choices yield similar results.

G.3 Details of Experimental Setup for Sec. 5.1

We use $\beta = 0.5$—the learning rate of the parameter "inner-loop" optimization problem, π_0 : `center` for all experiments of this section. We use vary one parameter of the problem while keeping all others fixed cf. Table 2. We use 50 randomly generated MDPs for each configuration and compute the mean and standard deviation shown in the plots.

Experiment	Alg/MDP parameter	Values
k `sweep`—Fig. 1`(Left)`	k	{1, 5, 10, 20, 30}
	b	5
	γ	0.95
	$\|\mathcal{A}\|$	10
	T	10
b `sweep`—Fig. 1`(Center-Left)`	k	100
	b	{5, 10, 20, 30, 40}
	γ	0.95
	$\|\mathcal{A}\|$	10
	T	10
γ `sweep`—Fig. 1`(Center-Right)`	k	30
	b	5
	γ	{0.98, 0.95, 0.9, 0.85}
	$\|\mathcal{A}\|$	10
	T	10
$\|\mathcal{A}\|$ `sweep`—Fig. 1`(Right)`	k	100
	b	5
	γ	0.95
	$\|\mathcal{A}\|$	{2, 5, 10, 15}
	T	20

Table 2: The parameters used for the optimization in Sec. 5.1.

H Supplementary Results for Sec. 5: Numerical Studies

H.1 Supplementary results for Sec. 5.1

This section presents additional results to those in Sec. 5.1. Fig. 4 shows the final performance and is analogous to Fig.1 in the main text. Fig. 5 shows the optimality gap while learning for T iterations.

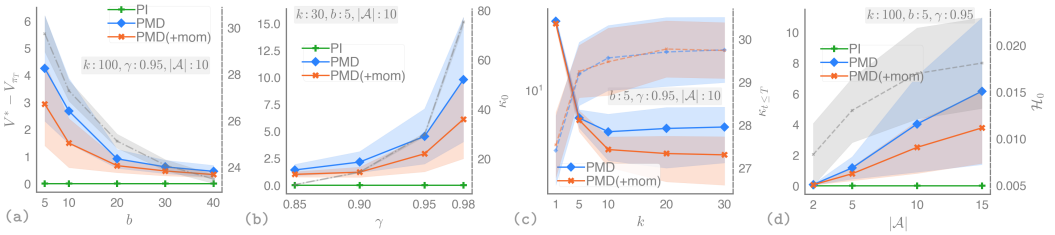

Fig. 4: The `left` y`-axis` shows the final optimality gap (regret) at timestep T (cf. Table 2) of the updates: `PI`, `PMD` and `PMD(+mom)`, after T iterations ($T = 10$ `(a-c)`, $T = 20$ `(d)`) relative to changing the hyperparameters: (a) b—the branching factor of the `Random MDP`, (b) γ—the discount factor, (c) k—the number of parameter updates, (d) $|\mathcal{A}|$—the number of actions. Shades denote standard deviation over 50 sampled MDPs. The `right` y`-axis` and dotted curves measure: `(a-b)`—the condition number κ_0, `(c)` the average condition number $\kappa_{t\leq T}$, `(d)` the entropy $\mathcal{H}_0$.

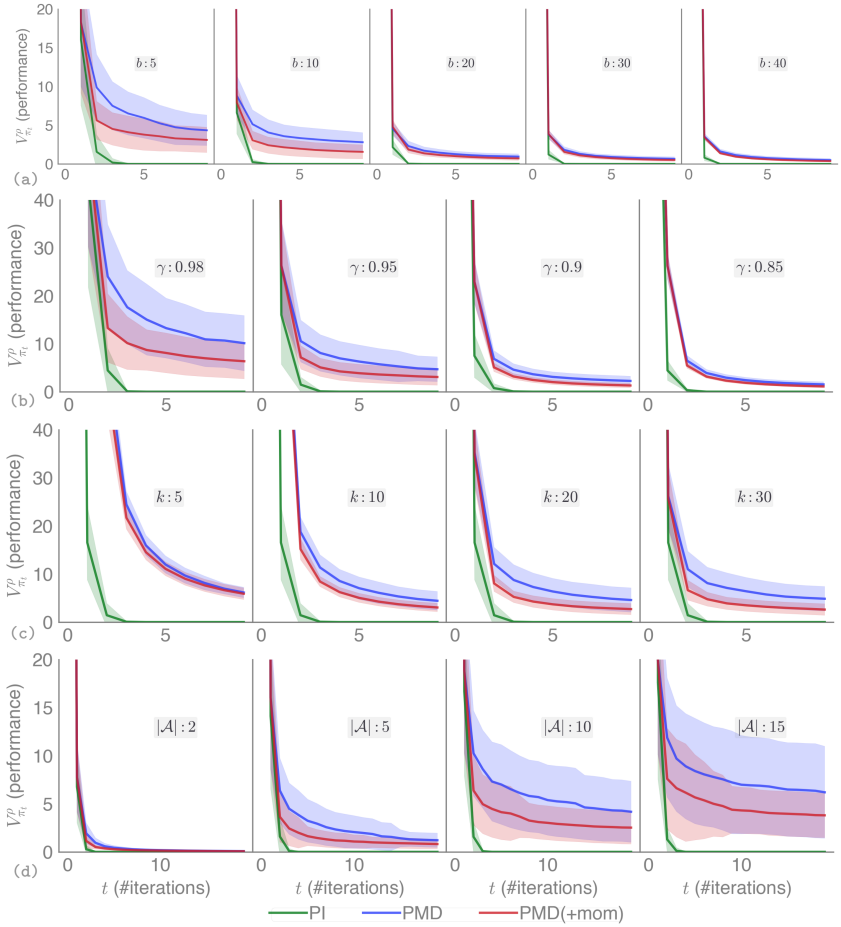

Fig. 5: The `left` y`-axis` shows the optimality gap (regret) of the updates: `PI`, `PMD` and `PMD(+mom)`, for T iterations ($T = 20$ final column, $T = 10$ rest of the columns) relative to changing the hyperparameters: (a) b—the branching factor of the `Random MDP`, (b) γ—the discount factor, (c) $|\mathcal{A}|$—the number of actions, (d) k—the number of parameter updates. Shades denote standard deviation over 50 sampled MDPs.

H.2 Supplementary results for Sec. 5.2

In this section we provide supplementary results that were omitted in the main body, related to the policy optimization dynamics of the functional acceleration methods introduced.

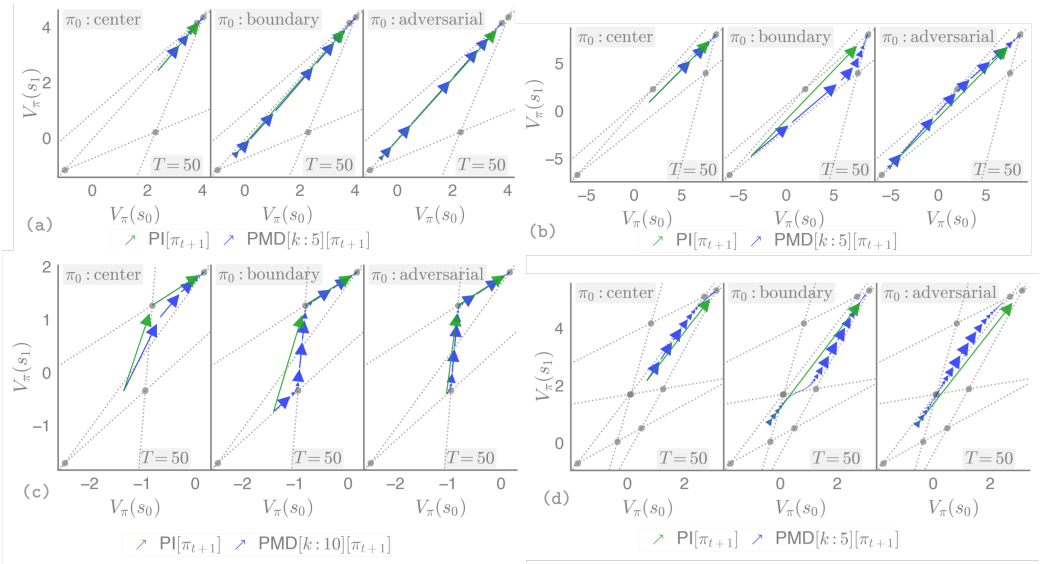

Fig. 6: Compares the policy optimization dynamics of `PMD` and `PI` on the value polytope of the different example MDPs in Sec. G.1: `(a)` example (ii), `(b)` example (iii), `(c)` example (i), `(d)` example (iv).

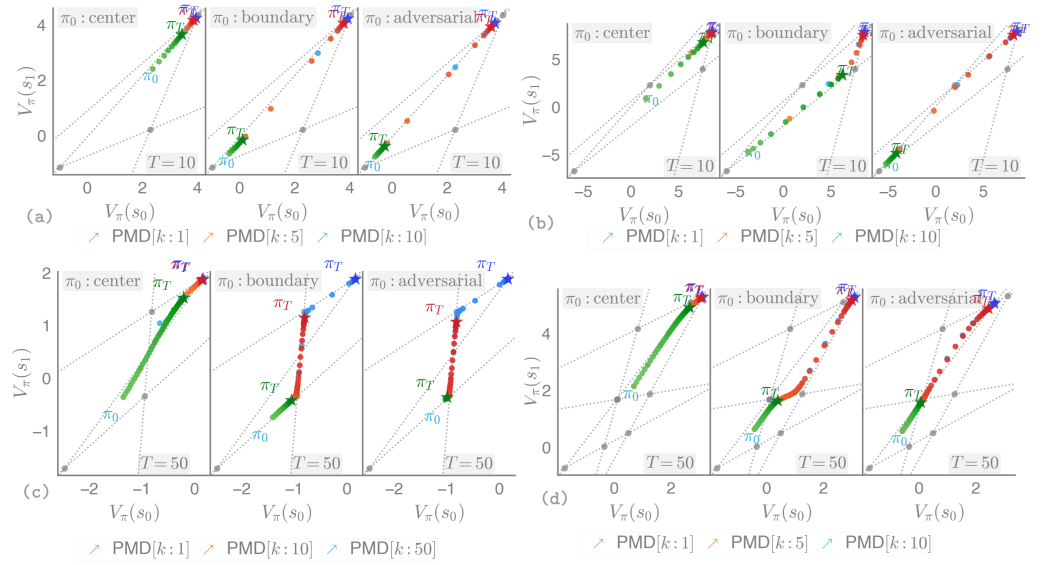

Fig. 7: Shows the policy optimization dynamics of `PMD` for different values of k on the value polytope of the different example MDPs in Sec. G.1: `(a)` example (ii), `(b)` example (iii), `(c)` example (i), `(d)` example (iv).

`PMD` In Fig. 6, we compare the optimization dynamics of `PMD` and `PI` for different MDPs. We observe the policy tends to move in a straight line between semi-deterministic policies (cf. Dadashi et al. (2019)) on the boundary of the polytope, and when it passes over an attractor point it can get delayed slowing down convergence. Fig. 7 shows the speed of convergence is governed by k which reflects the inner-loop optimizationn procedure. We again observe in Fig. 7 the accummulation points and long-escape attractor points of the optimization procedure.

`PMD(+loo)` In Fig. 8 we observe the dynamics of `PMD(+loo)` sometimes follow a different path through the polytope compared to `PMD` or `PI` (Fig. 6), as they are following a different ascent direction,

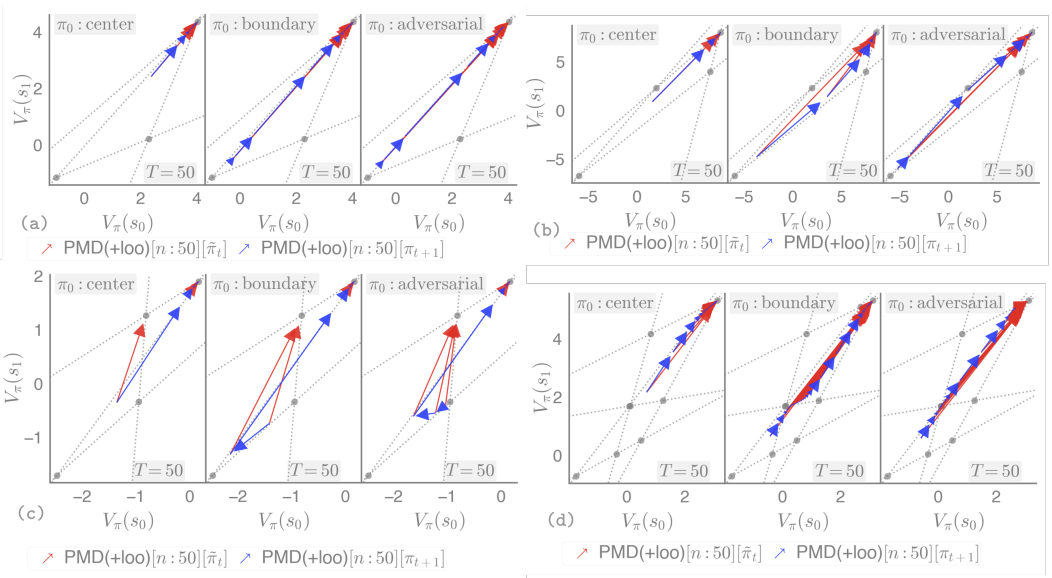

Fig. 8: Shows the policy optimization dynamics of `PMD(+loo)` on the value polytope of the different example MDPs in Sec. G.1: `(a)` example (ii), `(b)` example (iii), `(c)` example (i), `(d)` example (iv).

which may be more direct compared to that of `PI`. Compared to Fig. 7, in Fig. 9, we see less accumulation points, and more jumps, i.e. the policy improvement step returns policies at further distance apart.

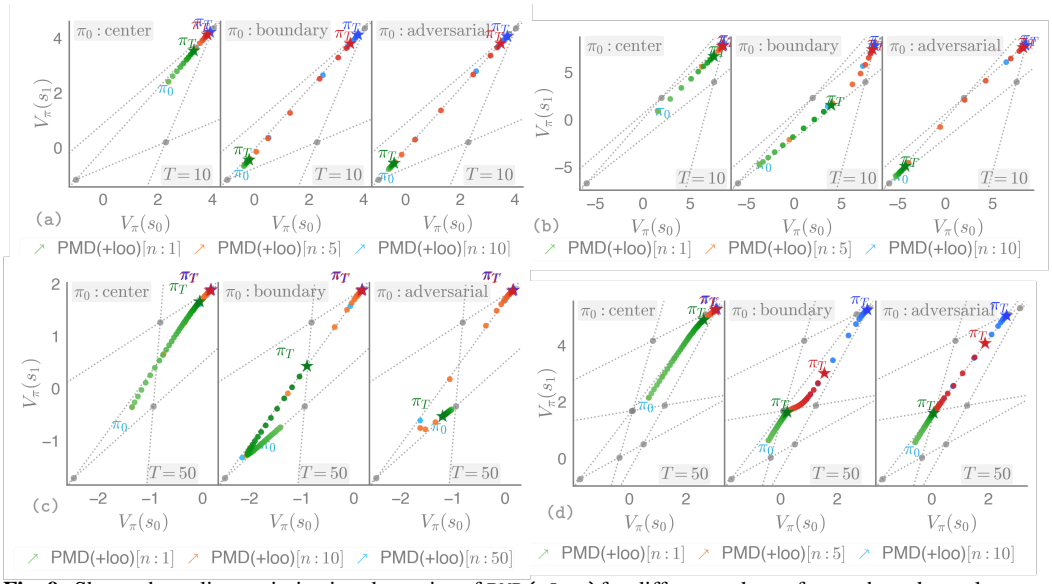

Fig. 9: Shows the policy optimization dynamics of `PMD(+loo)` for different values of n on the value polytope of the different example MDPs in Sec. G.1: `(a)` example (ii), `(b)` example (iii), `(c)` example (i), `(d)` example (iv).

`PMD(+ext)` In Fig. 10, we observe the dynamics of `PMD(+ext)`, specifically, the role of each the forward and backward steps. We may notice more speedup in regions where the ascent direction aligns over consecutive steps. We chose a value of n particularly to show the difference between the different kinds of steps. Then, in Fig. 11, we compare them against those of `PMD`, and find that in some cases, particularly those in which the problem is less complex, and the optimization surface less ill-conditioned, that using the next gradient is not better than the baseline `PMD`, which uses the current gradient.

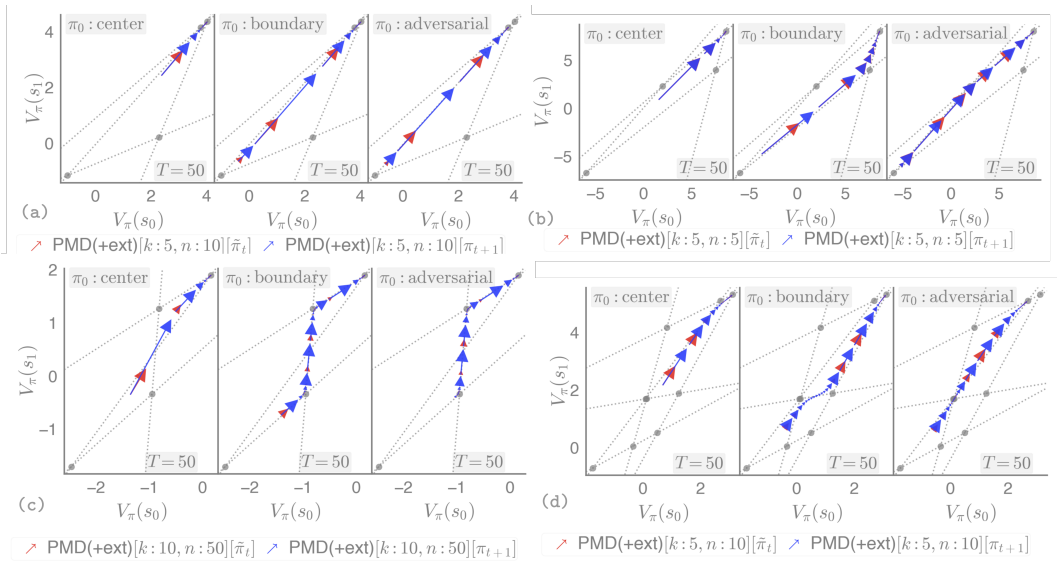

Fig. 10: Shows the policy optimization dynamics of `PMD(+ext)` on the value polytope of the different example MDPs in Sec. G.1: `(a)` example (ii), `(b)` example (iii), `(c)` example (i), `(d)` example (iv).

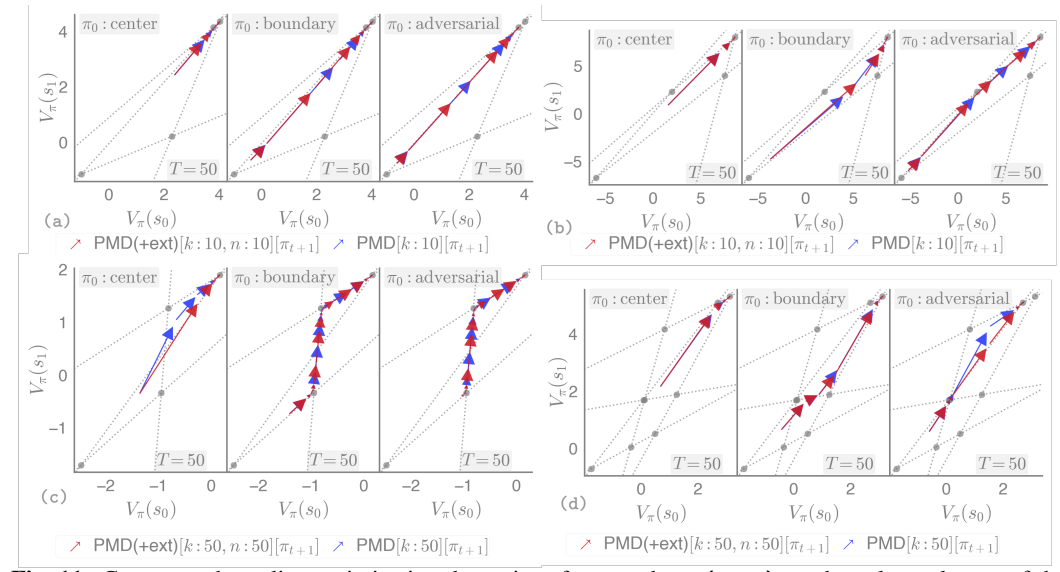

Fig. 11: Compares the policy optimization dynamics of `PMD` and `PMD(+ext)` on the value polytope of the different example MDPs in Sec. G.1: `(a)` example (ii), `(b)` example (iii), `(c)` example (i), `(d)` example (iv).

`PMD(+cor)` Fig. 12 shows that if the problem is too easy, and the optimal policy can be obtained in one greedy step, functional acceleration is unnecessary, since the next gradient may not particularly be better than the current one. Consequently, the correction of `PMD(+cor)` switches in the opposite direction, causing the optimization to decelerate. Whether this is a good idea or not may depend on the problem instance and the designer's goals. In many cases we do not want to ever reach stationarity annd may want to keep continually exploring. Compared to `PMD(+ext)`, Fig. 13 shows their dynamics are very similar.

`PMD(+lzc)` A similar story unfolds for `PMD(+lzc)` in Fig. 14, with the exception that the roles of the forward backward steps are reversed due to the lazy correction. Fig. 15 compares the optimization dynamics of `PMD(+cor)` and `PMD(+lzc)`, illustrating the impact of laziness. We may observe that despite their differences, the methods are quite similar in terms of acceleration. Notice that `PMD(+cor)` moves ahead in some parts of the optimization path but `PMD(+lzc)` catches up immediately after.

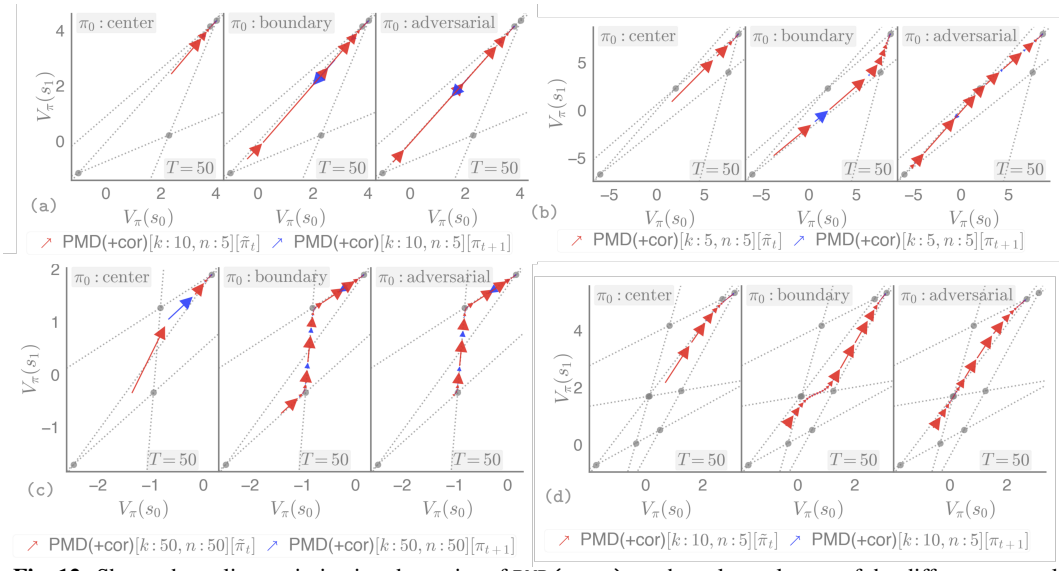

Fig. 12: Shows the policy optimization dynamics of `PMD(+cor)` on the value polytope of the different example MDPs in Sec. G.1: `(a)` example (ii), `(b)` example (iii), `(c)` example (i), `(d)` example (iv).

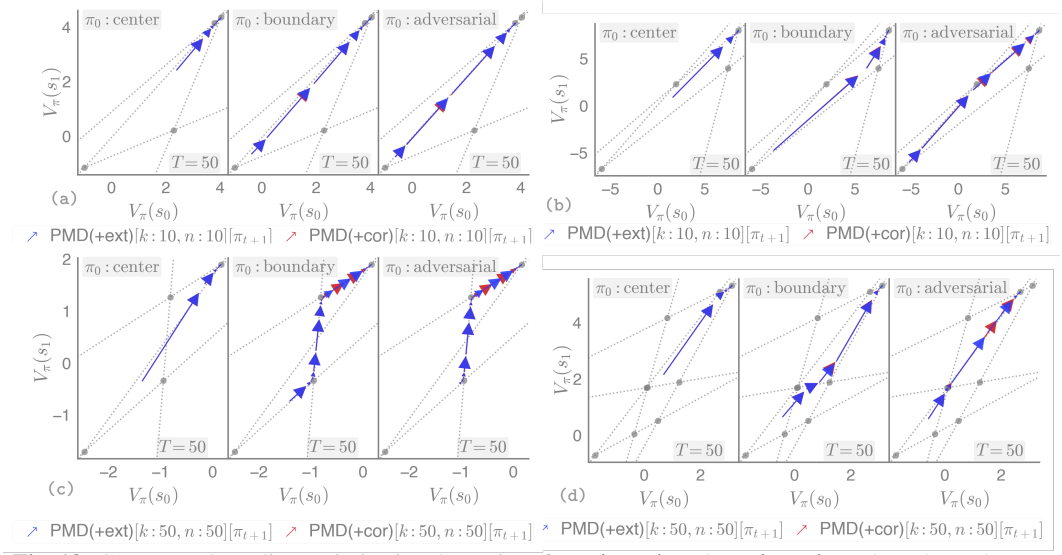

Fig. 13: Compares the policy optimization dynamics of `PMD(+ext)` and `PMD(+cor)` on the value polytope of the different example MDPs in Sec. G.1: `(a)` example (ii), `(b)` example (iii), `(c)` example (i), `(d)` example (iv).

`PMD(+mom)` Fig. 16 compares the policy dynamics of `PMD` and `PMD(+mom)` and shows acceleration of the latter in those directions of ascent that align over consecutive steps, and have ill-conditioned optimization surfaces. Fig. 17 compares the optimization dynamics of `PMD(+mom)` with those of `PMD(+lzc)`, which is consistent with the solution set inclusion property from Proposition4.

H.3 Supplementary results for Sec. 5.3

Fig. 18, 19, 20, 21 illustrate the variance over 50 optimization trajectories initialized from a random uniform distribution with mean 0 and standard deviation 1, for each of the example two-state MDPs described in Appendix G.1. We use the same `(controlled)` setting as in Sec.5.3, and vary the critic's error τ, and the policy approximation via k.

The most illustrative example is Fig. 18, since this setting presents the most ill-conditioned surface on which we can observe the impact of acceleration. We observe in `(a)` the instability of policy iteration

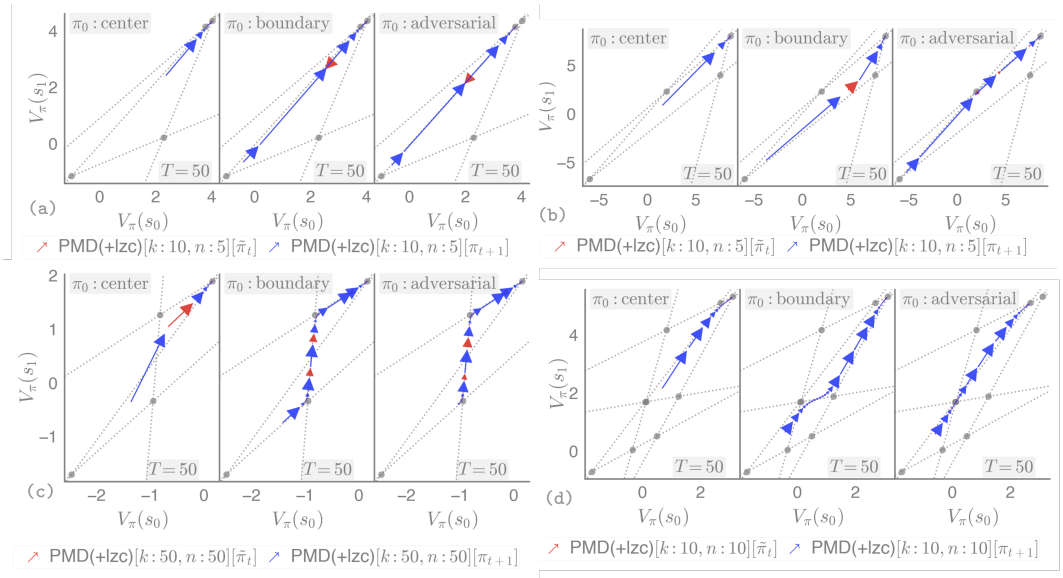

Fig. 14: Shows the policy optimization dynamics of `PMD(+lzc)` on the value polytope of the different example MDPs in Sec. G.1: `(a)` example (ii), `(b)` example (iii), `(c)` example (i), `(d)` example (iv).

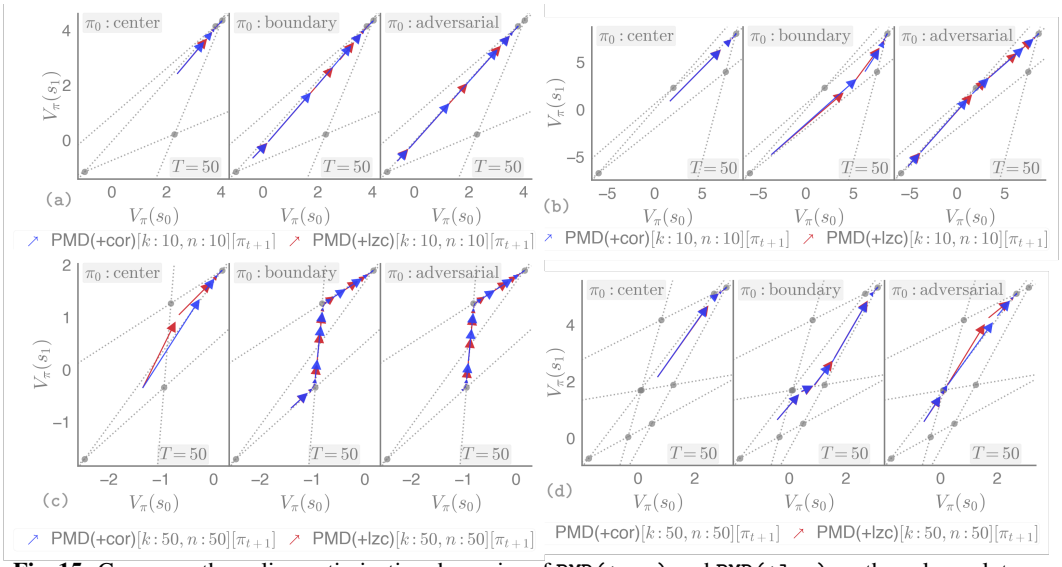

Fig. 15: Compares the policy optimization dynamics of `PMD(+cor)` and `PMD(+lzc)` on the value polytope of the different example MDPs in Sec. G.1: `(a)` example (ii), `(b)` example (iii), `(c)` example (i), `(d)` example (iv).

with an inexact critic. As the critic's error grows the policy iterates start to oscillate between the corners of the polytope.

`PMD` is better behaved for low k values and starts to exhibit behaviour similar to `PI` at larger k values `(c,d)`. We observe `PMD(+mom)` is more unstable than `PMD` when presented with high level of errors in the inexact critic `(f)`, and tends to stay more on the boundary of the polytope `(e)`, which is consistent with having larger values of the gradient due to added momentum.

In `(g,h)` the learning curves show the variance over trajectories stemming from the random initialization, the instability of `PI`, the relative improvement of `PMD(+mom)` over `PMD`, particularly striking for larger k consistent with the theory. We observe in `(h)` `PMD(+mom)` has more variance in the beginning, which may actually be desirable in terms of exploration, and that is achieves a similar performance to `PMD` at the end of the optimization.

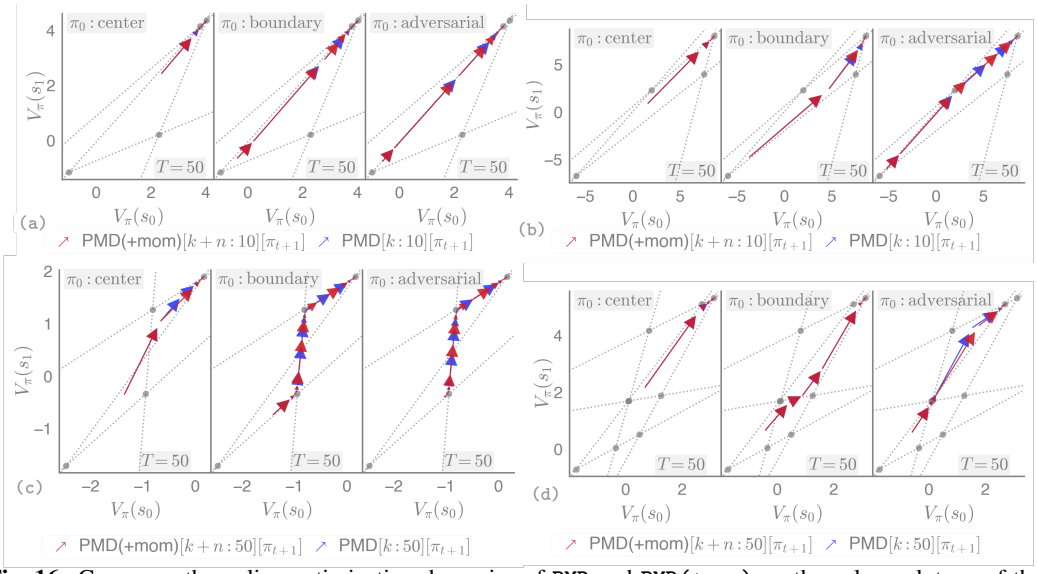

Fig. 16: Compares the policy optimization dynamics of `PMD` and `PMD(+mom)` on the value polytope of the different example MDPs in Sec. G.1: `(a)` example (ii), `(b)` example (iii), `(c)` example (i), `(d)` example (iv).

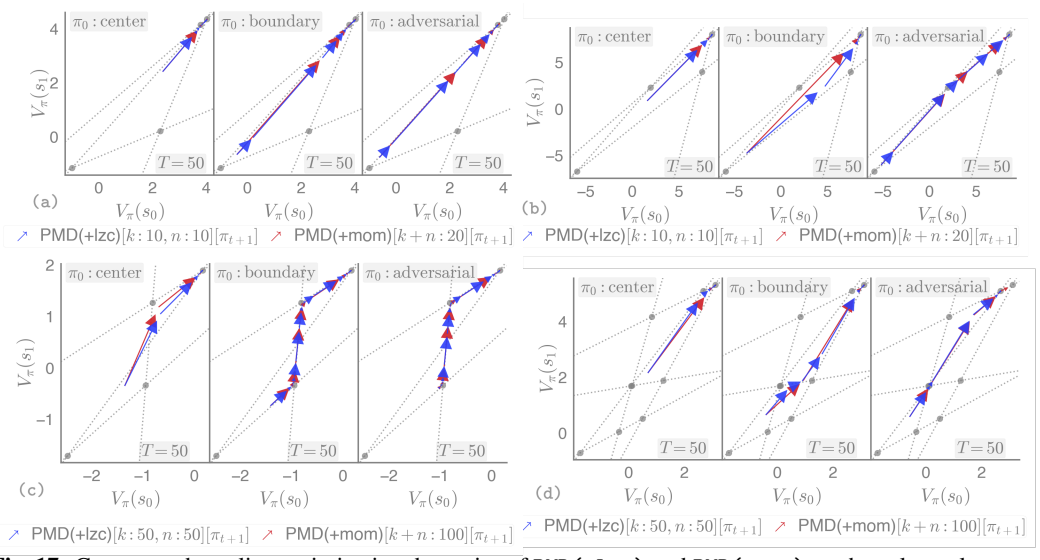

Fig. 17: Compares the policy optimization dynamics of `PMD(+lzc)` and `PMD(+mom)` on the value polytope of the different example MDPs in Sec. G.1: `(a)` example (ii), `(b)` example (iii), `(c)` example (i), `(d)` example (iv).

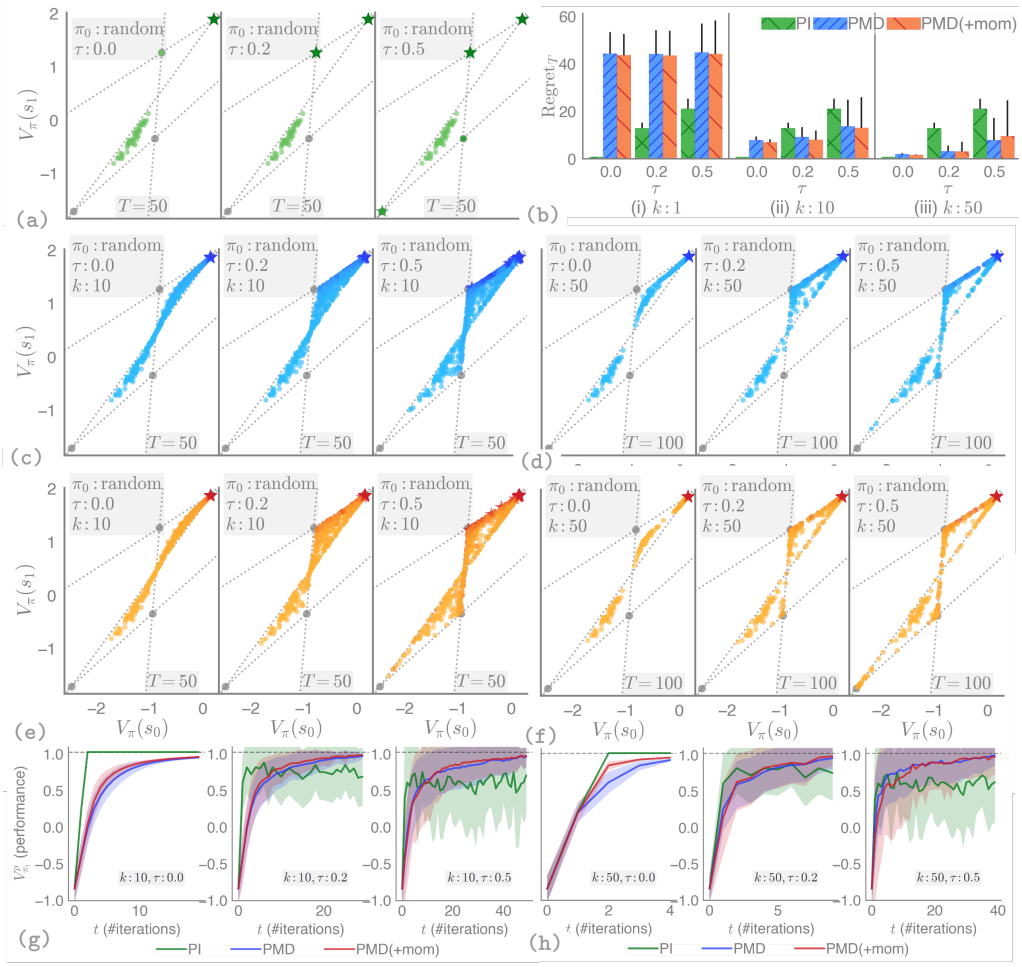

Fig. 18: Compares the statistics of the policy optimization dynamics of `PI`, `PMD` and `PMD(+mom)` subject to variance from random initialization (π_0 : `random_uniform(0,1)`, relative to the error in the inexact critic (τ), and over different levels of policy approximation (k). Results correspond to example (i) from Sec. G.1.

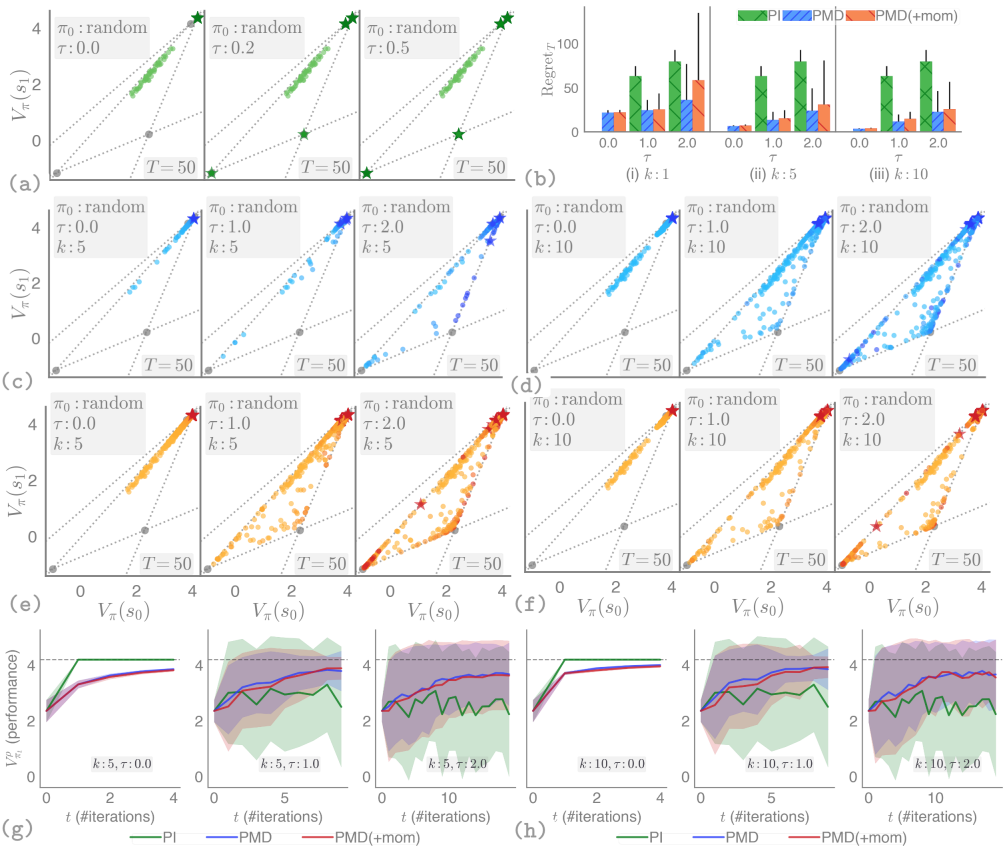

Fig. 19: Compares the statistics of the policy optimization dynamics of `PI`, `PMD` and `PMD(+mom)` subject to variance from random initialization (π_0 : `random_uniform(0,1)`, relative to the error in the inexact critic (τ), and over different levels of policy approximation (k). Results correspond to example (ii) from Sec. G.1.

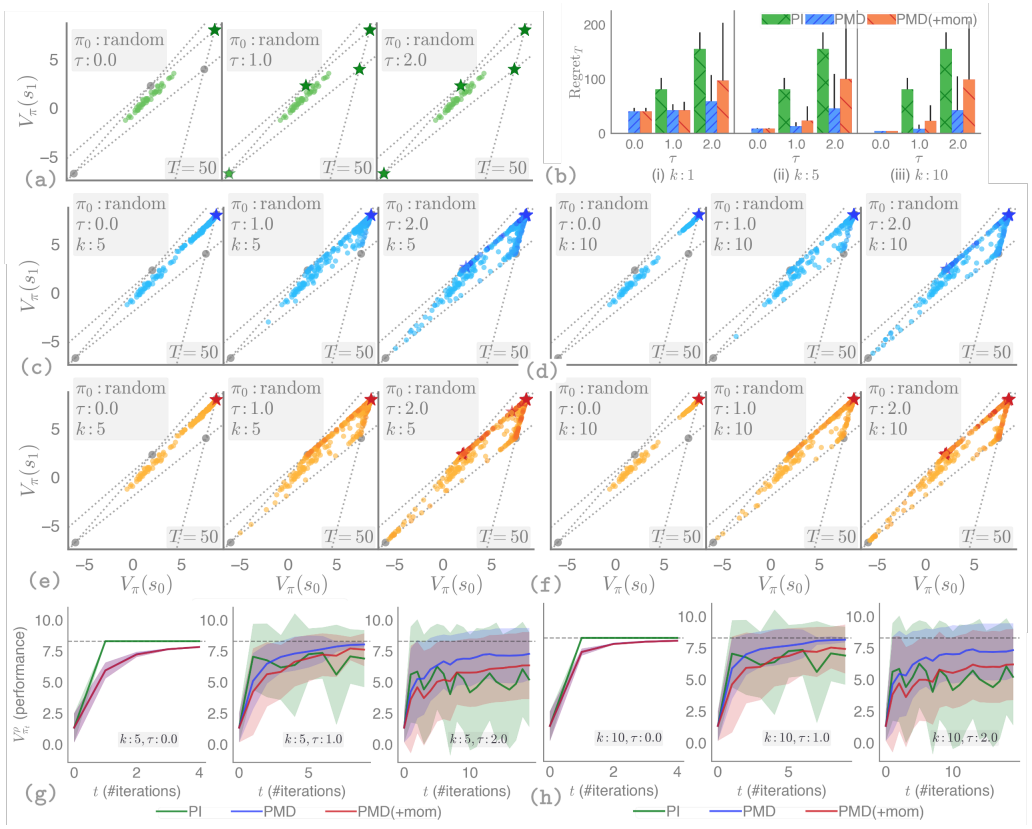

Fig. 20: Compares the statistics of the policy optimization dynamics of `PI`, `PMD` and `PMD(+mom)` subject to variance from random initialization (π_0 : `random_uniform(0,1)`, relative to the error in the inexact critic (τ), and over different levels of policy approximation (k). Results correspond to example (iii) from Sec. G.1.

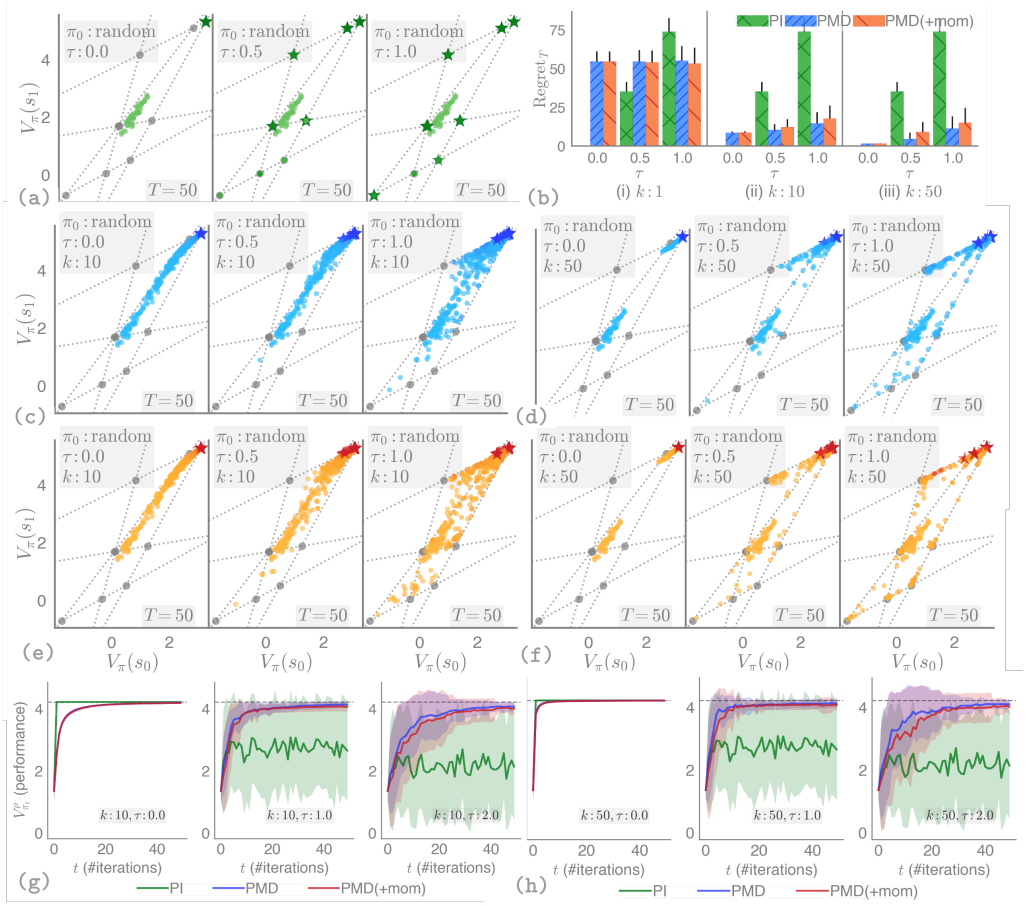

Fig. 21: Compares the statistics of the policy optimization dynamics of `PI`, `PMD` and `PMD(+mom)` subject to variance from random initialization (π_0 : `random_uniform(0,1)`, relative to the error in the inexact critic (τ), and over different levels of policy approximation (k). Results correspond to example (iv) from Sec. G.1.