Convergence and Sample Complexity of First-Order Methods for Agnostic Reinforcement Learning

Uri Sherman
Blavatnik School of Computer Science and AI
Tel Aviv University
urisherman@mail.tau.ac.il

Tomer Koren
Blavatnik School of Computer Science and AI
Tel Aviv University and Google Research
tkoren@tauex.tau.ac.il

Yishay Mansour
Blavatnik School of Computer Science and AI
Tel Aviv University and Google Research
mansour.yishay@gmail.com

Abstract

We study reinforcement learning (RL) in the agnostic policy learning setting, where the goal is to find a policy whose performance is competitive with the best policy in a given class of interest Π—crucially, without assuming that Π contains the optimal policy. We propose a general policy learning framework that reduces this problem to first-order optimization in a non-Euclidean space, leading to new algorithms as well as shedding light on the convergence properties of existing ones. Specifically, under the assumption that Π is convex and satisfies a variational gradient dominance (VGD) condition—an assumption known to be strictly weaker than more standard completeness and coverability conditions—we obtain sample complexity upper bounds for three policy learning algorithms: *(i)* Steepest Descent Policy Optimization, derived from a constrained steepest descent method for non-convex optimization; *(ii)* the classical Conservative Policy Iteration algorithm [Kakade and Langford, 2002] reinterpreted through the lens of the Frank-Wolfe method, which leads to improved convergence results; and *(iii)* an on-policy instantiation of the well-studied Policy Mirror Descent algorithm. Finally, we empirically evaluate the VGD condition across several standard environments, demonstrating the practical relevance of our key assumption.

1 Introduction

Policy Optimization (PO) algorithms are a class of methods in Reinforcement Learning (RL; Sutton and Barto, 2018, Mannor et al., 2022) in which an agent's policy is iteratively updated to minimize long-term cost, as defined by the environment's value functions. Modern applications of PO methods [e.g., Lillicrap, 2015, Schulman et al., 2015, Akkaya et al., 2019, Ouyang et al., 2022] often involve large-scale environments that lack well-defined structure, and by that require function approximation techniques in order to learn efficiently. Typically, PO algorithms represent the agent's policy using neural network models—commonly referred to as *actor* networks. Notably, these setups are inherently agnostic: the learner searches for an assignment of network parameters that is competitive with the best achievable under the model, without any guarantee that the optimal policy is expressible by the actor architecture.

Motivated by this, we consider the problem of *agnostic policy learning* in the general function approximation setup [Kakade, 2003, Krishnamurthy et al., 2025], where the learner is given optimization

18th European Workshop on Reinforcement Learning (EWRL 2025).

oracle access to a policy class Π and is required to find a policy that performs nearly as well as the best in-class policy. It is well known that Π-completeness and coverage conditions allow for sample efficient policy learning [Agarwal et al., 2019, 2021, Bhandari and Russo, 2024],[1] however, completeness implies realizability, and both conditions are generally deemed too strong to hold in practice. Furthermore, the extent to which they hold or not is hard to measure empirically.

In this work, we adopt instead the assumption that Π satisfies a variational gradient dominance condition [Agarwal et al., 2021, Xiao, 2022, Bhandari and Russo, 2024], which is known to be strictly weaker than completeness and coverage, and in particular, may accommodate non-realizable setups [Bhandari and Russo, 2024, Sherman et al., 2025] (see Fig. 1, and Appendix A for further details). Furthermore, the VGD parameters are to a degree measurable in practice, and appear better suited to characterize convergence of first-order policy learning algorithms. Indeed, the empirically observed parameters are reasonable compared to the theoretical, hard to measure ones associated completeness and coverage; and the VGD assumption pinpoints the precise properties required in convergence analysis under completeness and coverage.

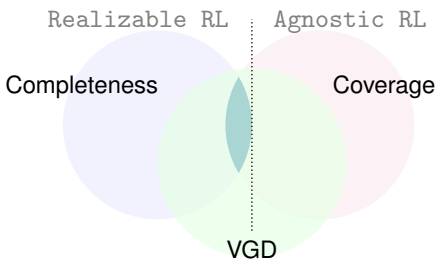

Figure 1: Completeness + coverage allows for sample efficient policy learning [Kakade and Langford, 2002]. These conditions imply in particular, realizability. VGD allows for sample efficient learning, and in particular accommodates agnostic (non-realizable) setups.

1.1 Our contributions

In this work, we make the following contributions.

Policy learning framework. We introduce a natural policy learning framework that reduces Agnostic RL to first order optimization in a constrained non-convex non-Euclidean setup. Consequently, we obtain practical, sample efficient[2] policy learning algorithms, and along the way also improvements in state-of-the-art iteration complexity upper bounds. Importantly, our framework and reduction are completely independent of the choice of policy class parametrization. In the function approximation policy learning setup, the primary way by which policies are produced is by constructing an objective function $\widehat{\Phi}_k : \Pi \to \mathbb{R}$, and invoking an optimization oracle to compute an approximate minimizer:

$$\pi^{k+1} \leftarrow \arg\min_{\pi \in \Pi} \widehat{\Phi}_k(\pi). \tag{1}$$

Roughly speaking, our reduction makes use of the policy gradient theorem [Sutton et al., 1999] in the direct parametrization case, along with an on-policy estimation scheme and a standard concentration argument to yield that with a suitable choice of $\widehat{\Phi}_k$, Eq. (1) produces a gradient step w.r.t. the value function in policy (i.e., state-action, functional) space. This, combined with the local-smoothness property of the value function [Sherman et al., 2025], implies the algorithm may be cast as a first-order method taking gradient steps on a smooth objective. Crucially, unlike Euclidean smoothness used in prior work [e.g., Agarwal et al., 2021], this leads to rates that are independent of the size of the state space. Furthermore, it is substantially different than operating in the parameter space of the policy parametrization [e.g., Mei et al., 2020, Yuan et al., 2022, Bhandari and Russo, 2024].

Non-Euclidean smooth constrained optimization. We highlight smooth constrained non-convex optimization in a non-Euclidean space as the principal setting to which agnostic RL reduces to. In this context, we provide novel analyses that, to our knowledge, have not appeared in the literature previously, including (i) a steepest descent method for smooth non-convex constrained optimization, and (ii) an analysis for an approximate Frank-Wolfe that holds for VGD objectives (a weaker condition compared to convexity). The constrained steepest descent method which we analyze here is a natural generalization of gradient descent to objectives smooth w.r.t. a non-Euclidean norm. Kelner et al.

[1] Roughly speaking, Π-completeness is defined as closure to policy improvement steps, and coverage as a constant upper bound on the worst case ratio between the initial state distribution and the optimal policy occupancy measure.

[2] By sample efficient, we refer to methods with convergence bounds that scale with the log-covering number of the policy class, but not with the cardinality of the state space.

Method	Rate	GP	Ag	NC	AOE
CPI [a] [Kakade and Langford, 2002]	$\frac{D_\infty}{\sqrt{K}} + \mathcal{E}(\Pi)\, D_\infty$	✓	✗	✓	✗
Log-linear NPG [b] [Agarwal et al., 2021]	$\frac{1}{\sqrt{K}} + \sqrt{\kappa\varepsilon + D_\infty \varepsilon_{\text{approx}}}$	✗	✗	✓	✓
Log-linear NPG [c] [Yuan et al., 2023]	$\left(1 - \frac{1}{D_\infty}\right)^K + D_\infty \sqrt{C_\nu(\varepsilon + \varepsilon_{\text{approx}})}$	✗	✗	✓	✓
PMD [d] [Alfano et al., 2023]	$(1 - \frac{1}{D_\infty})^K + D_\infty \sqrt{C_\nu\, \varepsilon_{\text{pmdc}}}$	✗	✗	✓	✓
PMD [Sherman et al., 2025]	$\frac{\nu^2}{K^{2/3}} + (\nu + K^{1/6})\, \varepsilon^{1/4} + \varepsilon_{\text{vgd}}$	✓	✓	✗	✓
SDPO **(This Work)**	$\frac{\nu^2}{K^{2/3}} + \nu K^{1/6} \sqrt{\varepsilon} + \varepsilon_{\text{vgd}}$	✓	✓	✗	✓
CPI [e] **(This Work)**	$\frac{\nu^2}{K} + \nu\, \varepsilon + \varepsilon_{\text{vgd}}$	✓	✓	✓	✗
DA-CPI **(This Work)**	$\frac{\nu^2}{K^{2/3}} + \nu^2 \varepsilon^{2/3} K^{2/3} + \varepsilon_{\text{vgd}}$	✓	✓	✗	✓

Table 1: Comparison of different policy optimization algorithms. **Rate:** Gives the suboptimality after K iterations, as a function of error terms. ε denotes the value fitting error under the relevant sampling distribution, and D_∞ the distribution mismatch coefficient. $\left\{\mathcal{E}(\Pi), \varepsilon_{\text{approx}}, \varepsilon_{\text{pmdc}}\right\}$ all measure some form of "completeness error"; refer to Appendix A for further details. **GP:** Whether the method applies for general parameterizations of Π; **Ag:** Whether it gives meaningful guarantees in an agnostic setting under the VGD assumption (replace $\nu \to D_\infty$ to compare with non-agnostic bounds, error floors are similar); **NC:** Whether it applies for Non-Convex Π; **AOE:** Whether it is actor-oracle efficient. **[a, b, c, d, e]:** [a] The dependence on the completeness error is given in [Scherrer and Geist, 2014]. [b] κ relates to an eigenvalue condition of the state-action features covariance under the learning distribution. [b+c] Both works also provide bounds in terms of a bias error which we do not include here. [c+d] These rates require geometrically increasing step sizes, for constant step sizes sub-linear rates exist. [e] Our version of CPI makes a different choice of step sizes.

[2014] appear to be the first to analyze the method for convex functions in the unconstrained setting, while Xiao [2022] provides an analysis for VGD objectives in the constrained Euclidean setup. Our work is the first to further extend the method to the constrained, non-Euclidean setup, using a non-standard notion of steepest descent direction w.r.t. a constrained set of potential gradient mappings. The optimization setup and our analyses naturally accommodate also *local smoothness* of the objective function, which is crucial for the interesting cases of the reduction mentioned in the preceding paragraph.

Iteration and sample complexity for policy learning algorithms. Combining the elements above, and assuming the policy class satisfies a VGD condition and is convex, we obtain upper bounds on the sample complexity of several algorithms within our proposed framework. In particular, we propose a sample efficient **Steepest Descent Policy Optimization (SDPO)** method, based on a constrained steepest descent method for non-convex, non-Euclidean optimization which we analyze in this work for the first time. We then revisit the classic **Conservative Policy Iteration** (CPI; Kakade and Langford, 2002) algorithm, and cast it as an instance of the well-known Frank-Wolfe [Frank and Wolfe, 1956] algorithm, leading to (i) improved iteration complexity, and (ii) a variant of CPI—Doubly Approximate CPI (DA-CPI)—which is sample efficient *and* more practical, as we explain shortly in the discussion that follows. Finally, we establish polynomial sample complexity for the well studied **Policy Mirror Descent** [Tomar et al., 2020, Xiao, 2022, Lan, 2023] algorithm. To the best of our knowledge, our work is the first to obtain sample complexity upper bounds for PMD that are independent of the policy class parametrization.

Table 1 gives a detailed comparison between several algorithms of interest. In particular, our SDPO bound provides a substantial improvement over PMD in the agnostic setting [Sherman et al., 2025], by obtaining $\sqrt{\varepsilon}$ error dependence rather than $\varepsilon^{1/4}$. When converting the result to a sample complexity upper bound this becomes significant. Furthermore, SDPO obtains better dependence on the action set cardinality A when applied with the L^1 action norm, thereby lifting one of the two barriers left in

Sherman et al. [2025] to obtain rates for large action spaces (rates that scale at most logarithmically with A). The same is true for DA-CPI, which also improves upon the guarantee of PMD in the agnostic setting in a similar fashion.

While our new bound for CPI provides an even sharper improvement, it is important to note that CPI in its original form is not as practical as the other algorithms we consider, in the following sense. Let us call a policy obtained by an invocation to the optimization oracle, such as π^{k+1} in Eq. (1), an *actor*. We say a policy learning algorithm is actor-oracle efficient (AOE) if the following two conditions hold. (i) The objective functions $\widehat{\Phi}_k$ given to the optimization oracle can be evaluated in time that is independent of the size of the state space, and polynomial in other problem parameters.[3] (ii) The actor space complexity—i.e., the maximal number of actors the algorithm requires to maintain in memory at any given time—is $O(1)$. Actor-oracle inefficient algorithms are generally not feasible (at least at the present time) for practical applications; for example, actor-memory linear in the number of iterations requires maintaining a prohibitively large amount of separate neural network models in memory. As we discuss further in Section 4.2, CPI requires linear actor memory, while PMD, SDPO, and DA-CPI are all actor efficient, requiring at most two actor models at any given time.

1.2 Related work

There is a rich line of work that studies the PMD algorithm [Agarwal et al., 2021, Xiao, 2022, Lan, 2023, Alfano et al., 2023, Ju and Lan, 2022], however these, including Sherman et al. [2025], either focus only on the optimization setup, or consider sample complexity subject to specific parametrization choices. The CPI algorithm was originally introduced by [Kakade and Langford, 2002], and its guarantees in terms of a completeness error was derived in [Scherrer and Geist, 2014] (see also Agarwal et al., 2019). Our work is directly inspired by Sherman et al. [2025], where the connection of PMD to a constrained non-Euclidean optimization setup was recently established. Here, we take a more problem-centric view of agnostic RL and establish a broader connection between policy learning and optimization. Due to space constraints, we defer additional discussion of related work to Appendix A.1.

2 Preliminaries

Discounted MDPs. A discounted MDP $\mathcal{M}$ is defined by a tuple $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathbb{P}, r, \gamma, \rho_0)$, where $\mathcal{S}$ denotes the state-space, $\mathcal{A}$ the action set, $\mathbb{P}\colon \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$ the transition dynamics, $r\colon \mathcal{S} \times \mathcal{A} \to [0, 1]$ the regret (i.e., cost) function, $0 < \gamma < 1$ the discount factor, and $\rho_0 \in \Delta(\mathcal{S})$ the initial state distribution. We assume the action set is finite with $A \coloneqq |\mathcal{A}|$, and identify $\mathbb{R}^A$ with $\mathbb{R}^{\mathcal{A}}$. We additionally assume, for clarity of exposition and in favor of simplified technical arguments, that the state space is finite with $S \coloneqq |\mathcal{S}|$, and identify $\mathbb{R}^{\mathcal{S}}$ with $\mathbb{R}^S$. We further denote the effective horizon by $H \coloneqq \frac{1}{1-\gamma}$. We emphasize that all our arguments may be extended to the infinite state-space setting with additional technical work. An agent interacting with the MDP is modeled by a policy $\pi\colon \mathcal{S} \to \Delta(\mathcal{A})$, for which we let $\pi_s \in \Delta(\mathcal{A}) \subset \mathbb{R}^A$ denote the action probability vector at s and $\pi_{s,a} \in [0, 1]$ denote the probability of taking action a at s. We denote the *value* and Q-function by:

$$V(\pi) \coloneqq \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \mid s_0 \sim \rho_0, \pi\right]; \quad Q^{\pi}_{s,a} \coloneqq \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \mid s_0 = s, a_0 = a, \pi\right].$$

We further denote the discounted state-occupancy measure of π by μ^{π}:

$$\mu^{\pi}(s) \coloneqq (1-\gamma) \sum_{t=0}^{\infty} \gamma^t \Pr(s_t = s \mid s_0 \sim \rho_0, \pi).$$

It is easily verified that $\mu^{\pi} \in \Delta(\mathcal{S})$ is indeed a state probability measure.

Learning objective. We consider the problem of learning an approximately optimal policy within a given policy class $\Pi \subset \Delta(\mathcal{A})^{\mathcal{S}}$:

$$\arg\min_{\pi \in \Pi} V(\pi). \tag{2}$$

[3] The intention is that for any fixed π, $\widehat{\Phi}_k(\pi)$ is computable in poly time (that is independent of the state space cardinality).

To avoid ambiguity, we denote the optimal value attainable by an in-class policy (a solution to Eq. (2)) by $V^\star(\Pi)$, and the optimal value attainable by any policy by $V^\star$:

$$V^\star(\Pi) \coloneqq \arg\min_{\pi^\star \in \Pi} V(\pi^\star); \quad V^\star \coloneqq \arg\min_{\pi^\star \in \Delta(\mathcal{A})^{\mathcal{S}}} V(\pi^\star). \tag{3}$$

Throughout this paper, we let $\pi^\star = \arg\min_{\pi \in \Pi} V(\pi)$, and $\mu^\star \coloneqq \mu^{\pi^\star}$. The policy class Π will always be clear from context. When we are in need to refer to the optimal policy / occupancy measure w.r.t. $\Pi_{\mathrm{all}} \coloneqq \Delta(\mathcal{A})^{\mathcal{S}}$, we will say so explicitly.

2.1 Problem setup

We consider agnostic policy learning in the standard offline optimization oracle model. Given an objective function $\phi\colon \Pi \to \mathbb{R}$ an approximate minimizer may be produced by invoking the oracle. We will use the following notation for approximate minimization. For any set $\mathcal{X}$ and objective $\phi\colon \mathcal{X} \to \mathbb{R}$, we denote the set of ϵ-approximate minimizers by:

$$\arg\min^{\epsilon}_{x \in \mathcal{X}} \{\phi(x)\} \coloneqq \left\{ x \in \mathcal{X} \mid \phi(x) \leq \min_{x'} \phi(x') + \epsilon \right\}. \tag{4}$$

When the agent decides to initiate a rollout episode in the environment, she starts at an initial state $s_0 \sim \rho_0$, then for $t = 0, \ldots,$ chooses action a_t given s_t, incurs $r(s_t, a_t)$, and transitions to $s_{t+1} \sim \mathbb{P}(\cdot|s_t, a_t)$, until she decides to terminate the episode. The sample complexity of a learning algorithm is the number of interaction time steps required to reach an ϵ-approximate optimal policy, i.e., a policy $\hat{\pi}$ such that $V(\hat{\pi}) - V^\star(\Pi) \leq \epsilon$. As mentioned in the introduction, our key assumption is that Π satisfies the following variational gradient dominance condition.

Definition 1 (Variational Gradient Dominance)**.** We say that Π satisfies a $(\nu, \varepsilon_{\mathrm{vgd}})$-variational gradient dominance (VGD) condition w.r.t. a value function $V\colon \Delta(\mathcal{A})^{\mathcal{S}} \to \mathbb{R}$, if there exist constants $\nu, \varepsilon_{\mathrm{vgd}} > 0$, such that for any policy $\pi \in \Pi$:

$$V(\pi) - \min_{\pi^\star \in \Pi} V(\pi^\star) \leq \nu \max_{\tilde{\pi} \in \Pi} \langle \nabla V(\pi), \pi - \tilde{\pi} \rangle + \varepsilon_{\mathrm{vgd}}. \tag{5}$$

We conclude this section by repeating notations used throughout.

$$\mu^k \coloneqq \mu^{\pi^k}, \quad Q^k \coloneqq Q^{\pi^k}, \quad S \coloneqq |\mathcal{S}|, \quad A \coloneqq |\mathcal{A}|, \quad H \coloneqq \frac{1}{1-\gamma}.$$

3 Policy learning via Non-Euclidean smooth constrained optimization

In this section, we present our policy learning framework, and explain the reduction to first order optimization in a constrained, smooth non-Euclidean optimization setup. The reduction consists of the following three ingredients;

(i) Agnostic policy learning is cast as a first-order optimization problem over the policy (sometimes referred to as "functional") space Π that, crucially—is constrained and exhibits non-Euclidean geometry.

(ii) Smoothness of the value function function is established w.r.t. a (non-Euclidean) norm that measures distance between policies in a manner that is independent of the size of the state space. The choice of norm can be global (e.g., $\|\cdot\|_{\infty,1}$), or a local norm induced by the on-policy occupancy measure.

(iii) By the policy gradient theorem and a standard uniform concentration argument, we have that a gradient step on the value function may be approximated through on-policy sampling.

In more detail, consider that a standard template for first order optimization in policy space may be framed as iterating through minimization problems of the form:

$$\pi^{k+1} \leftarrow \arg\min_{\pi \in \Pi} \left\langle \nabla V(\pi^k), \pi \right\rangle + \frac{1}{\eta} \mathfrak{D}_\Pi\left(\pi, \pi^k\right), \tag{6}$$

where $\pi^1 \in \Pi$ is a given initialization and $\eta > 0$ a learning rate. When Π is a convex set and $\mathfrak{D}_\Pi$ is a distance-like function that is compatible with the geometry of the objective function V —

e.g., informally, when V is smooth w.r.t. $\mathfrak{D}_\Pi$ — some form of convergence may be established. In what follows, by a direct application of the policy gradient theorem [Sutton et al., 1999] and the recently established local smoothness property [Sherman et al., 2025], we will demonstrate how policy learning may be reduced to first order optimization in the form of Eq. (6). Let π^k is the agent's policy on iteration k, and let μ^k, Q^k be it's occupancy measure and action-value function. By the policy gradient theorem [Sutton et al., 1999], we have that for any policy π:

$$\mathbb{E}_{s\sim\mu^k}\left[H\left\langle Q_s^k, \pi_s\right\rangle\right] = \left\langle \nabla V(\pi^k), \pi\right\rangle.$$

Thus, using a proper sampling mechanism while interacting with the environment, we may obtain a dataset $\mathcal{D}_k$ consisting of states $s \sim \mu^k := \mu^{\pi^k}$, and unbiased action-value estimates $\widehat{Q}_s^k$. This gives an empirical version of the linearization $\frac{H}{N}\sum_{s\in\mathcal{D}_k}\left\langle \widehat{Q}_s^k, \pi_s\right\rangle$ that concentrates about the true gradient as N grows. Conveniently, given a distance-like function $\mathfrak{D}\colon \mathbb{R}^{\mathcal{A}} \times \mathbb{R}^{\mathcal{A}} \to \mathbb{R}$ over the action space, taking an expectation w.r.t. the on-policy distribution gives a policy distance-like function $\mathfrak{D}_k(\pi, \pi') := \mathbb{E}_{s\sim\mu^k}\mathfrak{D}(\pi_s, \pi'_s)$ w.r.t. which the value function is smooth [Sherman et al., 2025]. Importantly, in order to enjoy this smoothness, we must incorporate ε_{ex}-greedy exploration. To that end, we define the exploratory verion of Π as follows:

$$\Pi^{\varepsilon_{\text{ex}}} := \left\{(1-\varepsilon_{\text{ex}})\pi + \varepsilon_{\text{ex}} u \mid \pi \in \Pi, u_{s,a} \equiv 1/A\ \forall s, a\right\} \tag{7}$$

With this in mind, we consider on-policy algorithms, all of which hinge on optimizing empirical surrogates to the full gradient objective function. Concretely, the update step in each algorithm is of the form:

$$\pi^{k+1} \leftarrow \operatorname*{arg\,min}_{\pi\in\Pi^{\varepsilon_{\text{ex}}}}\left\{\widehat{\Phi}_k(\pi) := \frac{H}{N}\sum_{s\in\mathcal{D}_k}\left\langle \widehat{Q}_s^k, \pi_s\right\rangle + \frac{1}{\eta}\mathfrak{D}\left(\pi_s, \pi_s^k\right)\right\}. \tag{8}$$

The preceding discussion implies that, when N is sufficiently large, Eq. (8) is an approximate version of Eq. (6) with a distance measure $\mathcal{D}_\Pi$ that is adapted to local-smoothness of the objective, and as such Eq. (8) is an instance of smooth non-convex optimization in a non-Euclidean space. Primarily, the algorithms we consider in Section 4 differ in the choice of $\mathfrak{D}$. For the formal definition of the optimization setup our reduction leads to, we refer the reader to Appendix C. Finally, we note that while our algorithms SDPO, DA-CPI, and PMD operate over the ε_{ex}-exploratory version of Π, their output policies may be transformed back into Π with negligible loss of the objective, hence they are in fact proper agnostic learning algorithms.

4 Policy learning algorithms

In this section, we present our policy learning algorithms in their idealized form. Given the discussion from Section 3, sample complexity of each algorithm follows through the same algorithmic template and argument; we provide the full details in Appendix E.

4.1 Steepest Descent Policy Optimization

In this section, we present our first algorithm, which we derive from a generalization of gradient descent to non-Euclidean norms. Given a differentiable objective $f\colon \mathbb{R}^d \to \mathbb{R}$, an unconstrained, Euclidean gradient descent step can be written as:

$$x^+ = x - \eta\nabla f(x) = \operatorname*{arg\,min}_{y\in\mathbb{R}^d}\langle \nabla f(x), y\rangle + \frac{1}{2\eta}\|y-x\|_2^2.$$

When the objective f is smooth w.r.t. $\|\cdot\|_2$ and the step size is chosen appropriately, it is guaranteed that the step decreases the objective value, which can by harnessed to obtain convergence to a stationary point in a non-convex setting. A natural generalization of the gradient descent step to accommodate non-Euclidean geometries consists of simply replacing the proximity term $\|y-x\|_2^2$ with any other norm $\|\cdot\|$:

$$x^+ = \operatorname*{arg\,min}_{y\in\mathbb{R}^d}\langle \nabla f(x), y\rangle + \frac{1}{2\eta}\|y-x\|^2.$$

Algorithm 1 Steepest Descent Policy Optimization (SDPO)

Input: $K \geq 1, \eta > 0, \varepsilon > 0, \varepsilon_{\text{ex}} > 0, \Pi \in \Delta(\mathcal{A})^{\mathcal{S}}$, and action norm $\|\cdot\|_\circ : \mathbb{R}^{\mathcal{A}} \to \mathbb{R}$.
Initialize $\pi^1 \in \Pi^{\varepsilon_{\text{ex}}}$
for $k = 1$ **to** K **do**
  Update $\pi^{k+1} \leftarrow \arg\min^{\varepsilon}_{\pi \in \Pi^{\varepsilon_{\text{ex}}}} \mathbb{E}_{s\sim\mu^k} \left[H \left\langle Q^k_s, \pi_s \right\rangle + \frac{1}{2\eta} \left\| \pi_s - \pi^k_s \right\|^2_\circ \right]$
end for
return $\hat{\pi} \coloneqq \pi^{K+1}$

As in the Euclidean case, when f is smooth w.r.t. $\|\cdot\|$ and the step size is chosen appropriately, the step decreases the objective value, which leads to convergence when applied iteratively over K steps. A relatively straightforward argument can be employed to establish $O(1/\sqrt{K})$ convergence to an approximate stationary point. The work of Kelner et al. [2014] appears to be the first to prove convergence of $O(1/K)$ assuming convexity of f, at least in this general form, as noted by the authors. In this work, we analyze for the first time the constrained version of the steepest descent method, which requires some care and extra machinery to cope simultaneously with constraints and the non-Euclidean nature of the updates. We obtain a $O(1/K)$ upper bound for VGD functions (a condition weaker than convexity) and $O(1/\sqrt{K})$ for general non-convex functions. To the best of our knowledge, this method was not considered by any prior work in the constrained setting, for any class of objective functions.

The iteration complexity given by Theorem 1 below, follows by our reduction explained in Section 3 (excluding the probabilistic part). The ε parameter passed to Algorithm 1 should be understood as the sum of the generalization error, originating in the noisy estimates of the objective, and optimization error, originating from the error of the optimization oracle.

Theorem 1. *Let Π be a convex policy class that satisfies $(\nu, \varepsilon_{\text{vgd}})$-VGD w.r.t. $\mathcal{M}$. Suppose that SDPO (Algorithm 1) is executed with the L^1 action norm $\|\cdot\|_1$. Then, after K iterations, with appropriately tuned η and ε_{ex}, the output of SDPO satisfies:*

$$V(\hat{\pi}) - V^\star(\Pi) = O\left(\frac{\nu^2 A H^4}{K^{2/3}} + \nu H^3 \sqrt{A} K^{1/6} \sqrt{\varepsilon} + \varepsilon_{\text{vgd}} \right).$$

As mentioned in the introduction, SDPO improves upon the previous guarantees of PMD [Sherman et al., 2025] in two regards; (i) dependence on the error $\varepsilon^{1/4} \to \varepsilon^{1/2}$, and (ii) dependence on A. Further, for the case of Euclidean action geometry, the SDPO analysis can be seen as a tighter analysis for PMD with Euclidean action-regularization case (note that the space the algorithms operate in is still non-Euclidean, only the action space is).

4.2 Conservative Policy Iteration

In this section, we present our results relating to the CPI algorithm (Kakade and Langford, 2002; see also Agarwal et al., 2019). We first consider CPI in its original form, presented in Algorithm 2. Kakade and Langford [2002] established, that when the step sizes η_k are chosen "greedily" so as to maximize the observed advantage gain, an $O(1/\varepsilon^2)$ iteration complexity follows, as long as the policy class is complete and the distribution mismatch is bounded (see our introduction and Appendix A).

Algorithm 2 Conservative Policy Iteration (CPI; Kakade and Langford, 2002)

input: Initial policy $\pi^1 \in \Pi$, Error tolerance $\varepsilon > 0$
for $k = 1, 2, \ldots,$ **do**
  Update $\tilde{\pi}^{k+1} \leftarrow \arg\min^{\varepsilon}_{\pi \in \Pi} \mathbb{E}_{s\sim\mu^k} \left\langle H Q^k_s, \pi_s - \pi^k_s \right\rangle$
  Set $\pi^{k+1} = (1 - \eta_k)\pi^k + \eta_k \tilde{\pi}^{k+1}$
end for

However, as it turns out, this is not the optimal step size choice. The following two observations imply that CPI is (an approximate version of) the Frank-Wolfe [Frank and Wolfe, 1956] algorithm applied in state-action space, with policies as the optimization variables and the value function as the objective.

Indeed, we have that (i) $\mathbb{E}_{s\sim\mu^k}\left\langle Q_s^k, \pi_s - \pi_s^k\right\rangle = \frac{1}{H}\left\langle \nabla V(\pi^k), \pi - \pi^k\right\rangle$, and further that (ii) the value function is $(2H^3)$-smooth w.r.t. the $\|\cdot\|_{\infty,1}$ norm — a property which we prove in Appendix D.3. Given (i) and (ii), it follows that convergence of CPI may be established through a standard FW analysis, where indeed, greedily choosing the step sizes gives $O(1/\varepsilon^2)$ iteration complexity, while the choice of $\eta_k \approx \frac{1}{k}$ gives $O(1/\varepsilon)$ iteration complexity.

Theorem 2. *Let Π be a policy class that satisfies $(\nu, \varepsilon_{\text{vgd}})$-VGD w.r.t. $\mathcal{M}$. Suppose that CPI (Algorithm 2) is executed with the step size choices $\eta_k = \frac{2\nu}{k+2}$ for $k = 1, \ldots, K$. Then, we have the guarantee that:*

$$V(\pi^K) - V^\star(\Pi) \leq \frac{8(2\nu^2+1)H^3}{K} + 2\nu\varepsilon + \varepsilon_{\text{vgd}}$$

As mentioned, completeness and coverage imply VGD (see Appendix A for the full details), hence Theorem 2 is a strict improvement over the classically known $O(1/\sqrt{K})$ guarantees established in Kakade and Langford [2002]. Also in Appendix A, we show our bounds subject to VGD subsume those with the completeness error $\mathcal{E}(\Pi)$ [Scherrer and Geist, 2014]. Finally, we note that in this version of the algorithm, convexity of Π is not required (see Scherrer and Geist, 2014 for additional discussion), nor does our analysis require it.

Doubly Approximate CPI: An actor-oracle efficient algorithm. In the function approximation setup, the convex combination step *cannot* be computed as is, and therefore CPI turns out as actor-oracle inefficient. Indeed, the algorihtm requires keeping all policies $\pi^1, \ldots, \pi^K$ throughout execution. A possible remedy for this issue is to approximate the convex combination step with a second oracle invocation, thereby allowing to dispose of actors computed in previous rounds. However, a-priori, extending the analysis is far from immediate, as the convex combination step needs to be approximated through samples and it is unclear how to control the propagation of errors in the analysis. Fortunately, the local-smoothness framework together with a local-norm based analysis of FW (which in itself is straightforward) allows to control on-policy estimation errors and arrive at a convergence rate upper bound.

Algorithm 3 Doubly-Approximate CPI (DA-CPI)

input: $\eta_1, \ldots, \eta_K > 0$; $\varepsilon > 0, \varepsilon_{\text{ex}} > 0$, action norm $\|\cdot\|_\circ$.
for $k = 1, \ldots, K$ **do**
  Update $\tilde{\pi}^{k+1} \leftarrow \arg\min^{\varepsilon}_{\pi\in\Pi^{\varepsilon_{\text{ex}}}} \mathbb{E}_{s\sim\mu^k}\left\langle HQ_s^k, \pi_s - \pi_s^k\right\rangle$
  Update $\pi^{k+1} \leftarrow \arg\min^{\varepsilon}_{\pi\in\Pi^{\varepsilon_{\text{ex}}}} \mathbb{E}_{s\sim\mu^k}\left\|\pi_s - ((1-\eta_k)\pi_s^k + \eta_k\tilde{\pi}_s^{k+1})\right\|_\circ^2$
end for
return $\hat{\pi} = \pi^{K+1}$

Theorem 3. *Let Π be a convex policy class that satisfies $(\nu, \varepsilon_{\text{vgd}})$-VGD w.r.t. $\mathcal{M}$. Suppose that DA-CPI (Algorithm 3) is executed with the L^1 action norm $\|\cdot\|_1$. Then, for an appropriate setting of $\eta_1, \ldots, \eta_K$ and ε_{ex}, we have that the DA-CPI output satisfies:*

$$V(\hat{\pi}) - V^\star(\Pi) = O\left(\nu^2 AH^3\left(\frac{1}{K^{2/3}} + \varepsilon^{1/3} + \varepsilon^{2/3}K^{2/3}\right)\right).$$

4.3 Policy Mirror Descent

Convergence of PMD in the optimization setting similar to the one we consider here was recently established in Sherman et al. [2025]. With moderate additional work, we obtain the following iteration complexity upper bound that may be directly translated to a sample complexity guarantee. Since there are no changes in the algorithm, we defer its presentation to Appendix D.4.

Theorem 4. *Let Π be a convex policy class that satisfies $(\nu, \varepsilon_{\text{vgd}})$-VGD w.r.t. $\mathcal{M}$. Suppose that PMD (see Algorithm 6 in Appendix D.4) is executed with the L^2 action regularizer. Then, with an appropriate tuning of $\eta, \varepsilon_{\text{ex}}$, we have that the output of PMD satisfies:*

$$V(\hat{\pi}) - V^\star(\Pi) = O\left(\frac{\nu^2 A^{3/2}H^3}{K^{2/3}} + \left(\nu + H^2AK^{1/6}\right)\varepsilon^{1/4}\right).$$

Theorem 4 above, along with our reduction (see Section 3 and Appendix E), to our best knowledge lead to the first sample complexity upper bounds for PMD in the agnostic setting, which are completely independent of the policy class parametrization.

5 Experimental evaluation of the VGD condition

In this section, we present proof-of-concept experiments (Fig. 2), demonstrating the empirically observed parameters of L^2-SDPO (equivalently, L^2-PMD) executed in four environments: Cartpole-v1 and Acrobot-v1 [Brockman et al., 2016], and SpaceInvaders-MinAtar and Breakout-MinAtar [Young and Tian, 2019]. Code was written on top of the Gymnax framework [Lange, 2022], and parts of it were based off purejaxrl [Lu et al., 2022].

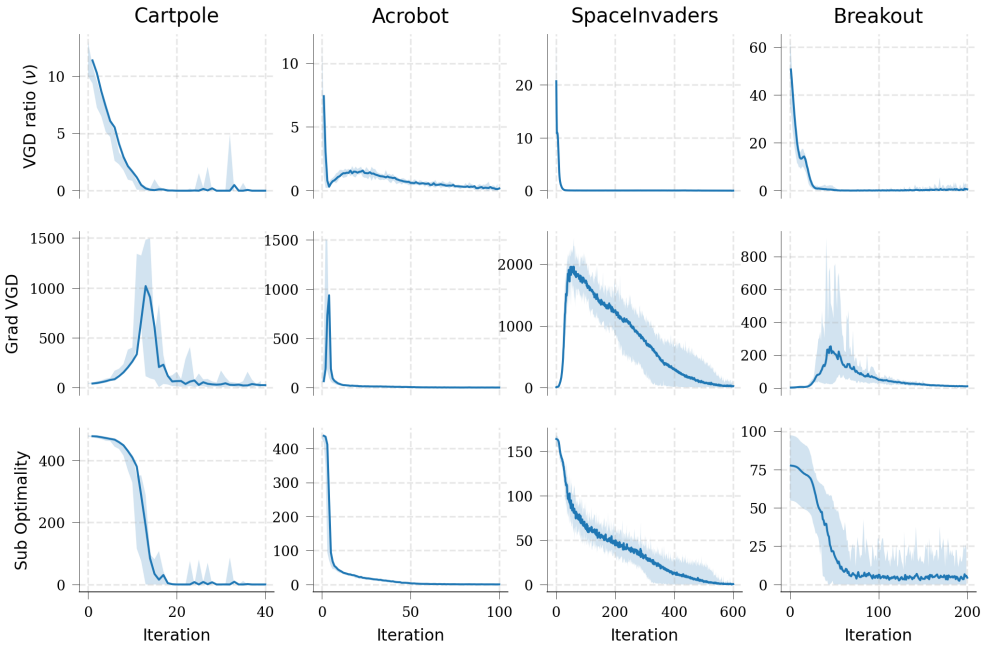

Figure 2: Training plots for all environments. In each experiment, a single set of minimally tuned hyper-parameters was run with 10 difference seeds. Error bars indicate maximum and minimum values. **VGD Ratio (ν):** An estimate of the ν parameter observed at each iteration $k \in [K]$ of the algorithm. **Grad VGD:** An estimate of $\max_{\tilde{\pi}\in\Pi} \left\langle \nabla V(\pi^k), \pi^k - \tilde{\pi} \right\rangle$. **Sub Optimality:** Sub optimality of iteration k w.r.t. the minimum value the algorithm converged to. This should be interpreted as $V(\pi^k) - (V^\star(\Pi) + \varepsilon_{\text{vgd}})$.

As can be seen in Fig. 2, the estimates of the ν-VGD parameter coefficient remain moderate throughout execution, and in fact decrease to around 1 or below as the algorithm approaches convergence.

We maintain two neural network models in the algorithm implementation, one for the original SDPO actor model, the other (a "VGD actor") we use to estimate the VGD parameter. In each iteration, we compute the SDPO step to obtain π^{k+1}, and in addition optimize the advantage function with the VGD actor in order to estimate $\max_{\tilde{\pi}\in\Pi} \left\langle \nabla V(\pi^k), \pi^k - \tilde{\pi} \right\rangle$. Since the algorithm may converge to a local optimal (which the ε_{vgd} parameter accounts for) we take the minimum value of each execution as the error floor $V^\star(\Pi) + \varepsilon_{\text{vgd}}$, and define the sub optimality as $V(\pi^k) - \left(V^\star(\Pi) + \varepsilon_{\text{vgd}}\right)$. We then report an estimate of ν by computing:

$$\frac{V(\pi^k) - \left(V^\star(\Pi) + \varepsilon_{\text{vgd}}\right)}{\max_{\tilde{\pi}\in\Pi} \left\langle \nabla V(\pi^k), \pi^k - \tilde{\pi} \right\rangle} = \nu_k. \tag{9}$$

In practice, for Cartpole and Acrobot the algorithm always converged to the global minimum; local optima only plays a role in the two more challenging environments. Given we believe the error floor, our reported ν_k are overestimates of the true parameter at π^k, as we find an actual policy $\tilde{\pi}$ that gives an upper bound on the LHS of Eq. (9). We provide further details on the experimental setup in Appendix F.

Acknowledgements

The authors would like to thank Tal Lancewicki and Alon Cohen for fruitful discussions and for their participation in the initial ideation process of this work. This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation program (grant agreements No. 101078075; 882396). Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Research Council. Neither the European Union nor the granting authority can be held responsible for them. This work received additional support from the Israel Science Foundation (ISF, grant numbers 3174/23; 1357/24), and a grant from the Tel Aviv University Center for AI and Data Science (TAD). This work was partially supported by the Deutsch Foundation.

References

A. Agarwal, N. Jiang, S. M. Kakade, and W. Sun. Reinforcement learning: Theory and algorithms. *CS Dept., UW Seattle, Seattle, WA, USA, Tech. Rep*, 32:96, 2019.

A. Agarwal, S. Kakade, A. Krishnamurthy, and W. Sun. Flambe: Structural complexity and representation learning of low rank mdps. *Advances in neural information processing systems*, 33: 20095–20107, 2020.

A. Agarwal, S. M. Kakade, J. D. Lee, and G. Mahajan. On the theory of policy gradient methods: Optimality, approximation, and distribution shift. *Journal of Machine Learning Research*, 22(98): 1–76, 2021.

I. Akkaya, M. Andrychowicz, M. Chociej, M. Litwin, B. McGrew, A. Petron, A. Paino, M. Plappert, G. Powell, R. Ribas, et al. Solving rubik's cube with a robot hand. *arXiv preprint arXiv:1910.07113*, 2019.

C. Alfano and P. Rebeschini. Linear convergence for natural policy gradient with log-linear policy parametrization. *arXiv preprint arXiv:2209.15382*, 2022.

C. Alfano, R. Yuan, and P. Rebeschini. A novel framework for policy mirror descent with general parameterization and linear convergence. *Advances in Neural Information Processing Systems*, 36: 30681–30725, 2023.

J. Bagnell, S. M. Kakade, J. Schneider, and A. Ng. Policy search by dynamic programming. *Advances in neural information processing systems*, 16, 2003.

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

D. Bertsekas and J. N. Tsitsiklis. *Neuro-dynamic programming*. Athena Scientific, 1996.

J. Bhandari and D. Russo. Global optimality guarantees for policy gradient methods. *Operations Research*, 2024.

G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba. Openai gym. *arXiv preprint arXiv:1606.01540*, 2016.

J. Chen and N. Jiang. Information-theoretic considerations in batch reinforcement learning. In *International Conference on Machine Learning*, pages 1042–1051. PMLR, 2019.

K. Dong, J. Peng, Y. Wang, and Y. Zhou. Root-n-regret for learning in markov decision processes with function approximation and low bellman rank. In *Conference on Learning Theory*, pages 1554–1557. PMLR, 2020.

S. Du, S. Kakade, J. Lee, S. Lovett, G. Mahajan, W. Sun, and R. Wang. Bilinear classes: A structural framework for provable generalization in rl. In *International Conference on Machine Learning*, pages 2826–2836. PMLR, 2021.

E. Even-Dar, S. M. Kakade, and Y. Mansour. Online markov decision processes. *Mathematics of Operations Research*, 34(3):726–736, 2009.

M. Frank and P. Wolfe. An algorithm for quadratic programming. *Naval research logistics quarterly*, 3(1-2):95–110, 1956.

M. Geist, B. Scherrer, and O. Pietquin. A theory of regularized markov decision processes. In *International Conference on Machine Learning*, pages 2160–2169. PMLR, 2019.

J. Grudzien, C. A. S. De Witt, and J. Foerster. Mirror learning: A unifying framework of policy optimisation. In *International Conference on Machine Learning*, pages 7825–7844. PMLR, 2022.

Z. Jia, G. Li, A. Rakhlin, A. Sekhari, and N. Srebro. When is agnostic reinforcement learning statistically tractable? *Advances in Neural Information Processing Systems*, 36:27820–27879, 2023.

N. Jiang, A. Krishnamurthy, A. Agarwal, J. Langford, and R. E. Schapire. Contextual decision processes with low bellman rank are pac-learnable. In *International Conference on Machine Learning*, pages 1704–1713. PMLR, 2017.

C. Jin, Z. Yang, Z. Wang, and M. I. Jordan. Provably efficient reinforcement learning with linear function approximation. In *Conference on Learning Theory*, pages 2137–2143. PMLR, 2020.

C. Jin, Q. Liu, and S. Miryoosefi. Bellman eluder dimension: New rich classes of rl problems, and sample-efficient algorithms. *Advances in neural information processing systems*, 34:13406–13418, 2021.

E. Johnson, C. Pike-Burke, and P. Rebeschini. Optimal convergence rate for exact policy mirror descent in discounted markov decision processes. *Advances in Neural Information Processing Systems*, 36:76496–76524, 2023.

C. Ju and G. Lan. Policy optimization over general state and action spaces. *arXiv preprint arXiv:2211.16715*, 2022.

S. Kakade and J. Langford. Approximately optimal approximate reinforcement learning. In *Proceedings of the Nineteenth International Conference on Machine Learning*, pages 267–274, 2002.

S. M. Kakade. A natural policy gradient. *Advances in neural information processing systems*, 14, 2001.

S. M. Kakade. *On the sample complexity of reinforcement learning*. University of London, University College London (United Kingdom), 2003.

J. A. Kelner, Y. T. Lee, L. Orecchia, and A. Sidford. An almost-linear-time algorithm for approximate max flow in undirected graphs, and its multicommodity generalizations. In *Proceedings of the twenty-fifth annual ACM-SIAM symposium on Discrete algorithms*, pages 217–226. SIAM, 2014.

A. Krishnamurthy, A. Agarwal, and J. Langford. Pac reinforcement learning with rich observations. *Advances in Neural Information Processing Systems*, 29, 2016.

A. Krishnamurthy, G. Li, and A. Sekhari. The role of environment access in agnostic reinforcement learning. *arXiv preprint arXiv:2504.05405*, 2025.

G. Lan. Policy mirror descent for reinforcement learning: Linear convergence, new sampling complexity, and generalized problem classes. *Mathematical programming*, 198(1):1059–1106, 2023.

R. T. Lange. gymnax: A JAX-based reinforcement learning environment library, 2022. URL `http://github.com/RobertTLange/gymnax`.

T. Lillicrap. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*, 2015.

C. Lu, J. Kuba, A. Letcher, L. Metz, C. Schroeder de Witt, and J. Foerster. Discovered policy optimisation. *Advances in Neural Information Processing Systems*, 35:16455–16468, 2022.

S. Mannor, Y. Mansour, and A. Tamar. *Reinforcement Learning: Foundations*. -, 2022. URL `https://sites.google.com/view/rlfoundations/home`.

J. Mei, C. Xiao, C. Szepesvari, and D. Schuurmans. On the global convergence rates of softmax policy gradient methods. In *International conference on machine learning*, pages 6820–6829. PMLR, 2020.

J. Mei, Y. Gao, B. Dai, C. Szepesvari, and D. Schuurmans. Leveraging non-uniformity in first-order non-convex optimization. In *International Conference on Machine Learning*, pages 7555–7564. PMLR, 2021.

S. Mu and D. Klabjan. On the second-order convergence of biased policy gradient algorithms. In *Forty-first International Conference on Machine Learning*, 2024.

R. Munos. Error bounds for approximate policy iteration. In *ICML*, volume 3, pages 560–567. Citeseer, 2003.

R. Munos. Error bounds for approximate value iteration. In *Proceedings of the National Conference on Artificial Intelligence*, volume 20, page 1006. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2005.

A. S. Nemirovskij and D. B. Yudin. Problem complexity and method efficiency in optimization, 1983.

L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022.

M. L. Puterman. Markov decision processes: Discrete stochastic dynamic programming, 1994.

B. Scherrer. Approximate policy iteration schemes: A comparison. In *International Conference on Machine Learning*, pages 1314–1322. PMLR, 2014.

B. Scherrer and M. Geist. Local policy search in a convex space and conservative policy iteration as boosted policy search. In *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2014*, pages 35–50. Springer, 2014.

J. Schulman, P. Moritz, S. Levine, M. Jordan, and P. Abbeel. High-dimensional continuous control using generalized advantage estimation. *arXiv preprint arXiv:1506.02438*, 2015.

A. Sekhari, C. Dann, M. Mohri, Y. Mansour, and K. Sridharan. Agnostic reinforcement learning with low-rank mdps and rich observations. *Advances in Neural Information Processing Systems*, 34: 19033–19045, 2021.

U. Sherman, T. Koren, and Y. Mansour. Convergence of policy mirror descent beyond compatible function approximation. *arXiv preprint arXiv:2502.11033*, 2025.

R. S. Sutton and A. G. Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

R. S. Sutton, D. McAllester, S. Singh, and Y. Mansour. Policy gradient methods for reinforcement learning with function approximation. *Advances in neural information processing systems*, 12, 1999.

M. Tomar, L. Shani, Y. Efroni, and M. Ghavamzadeh. Mirror descent policy optimization. *arXiv preprint arXiv:2005.09814*, 2020.

R. Vershynin. *High-dimensional probability: An introduction with applications in data science*, volume 47. Cambridge university press, 2018.

L. Xiao. On the convergence rates of policy gradient methods. *Journal of Machine Learning Research*, 23(282):1–36, 2022.

L. Yang and M. Wang. Sample-optimal parametric q-learning using linearly additive features. In *International Conference on Machine Learning*, pages 6995–7004. PMLR, 2019.

L. Yang and M. Wang. Reinforcement learning in feature space: Matrix bandit, kernels, and regret bound. In *International Conference on Machine Learning*, pages 10746–10756. PMLR, 2020.

K. Young and T. Tian. Minatar: An atari-inspired testbed for thorough and reproducible reinforcement learning experiments. *arXiv preprint arXiv:1903.03176*, 2019.

R. Yuan, R. M. Gower, and A. Lazaric. A general sample complexity analysis of vanilla policy gradient. In *International Conference on Artificial Intelligence and Statistics*, pages 3332–3380. PMLR, 2022.

R. Yuan, S. S. Du, R. M. Gower, A. Lazaric, and L. Xiao. Linear convergence of natural policy gradient methods with log-linear policies. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023.

W. Zhan, S. Cen, B. Huang, Y. Chen, J. D. Lee, and Y. Chi. Policy mirror descent for regularized reinforcement learning: A generalized framework with linear convergence. *SIAM Journal on Optimization*, 33(2):1061–1091, 2023.

K. Zhang, A. Koppel, H. Zhu, and T. Basar. Global convergence of policy gradient methods to (almost) locally optimal policies. *SIAM Journal on Control and Optimization*, 58(6):3586–3612, 2020.

A Comparison with prior art

In this section, we provide additional details regarding related work and relation between different assumptions mentioned in the introduction. We begin with formal definitions for Π-*completeness* and *coverage*. Completeness is also sometimes referred to as closure [Bhandari and Russo, 2024, Sherman et al., 2025], and coverage is a general term we use here to refer to bounded distribution-mismatch coefficient [Agarwal et al., 2019].

Definition 2 (Completness)**.** We say a policy class Π is complete if for any $\pi \in \Pi$, there exists a policy $\pi^+ \in \Pi$ such that for all $s \in \mathcal{S}$, $\pi^+_s = e_a$, where $a \in \arg\max_{a\in\mathcal{A}} Q^\pi_{s,a}$. In words, Π contains a policy π^+ that acts greedily w.r.t. the Q-function of π.

Definition 3 (Coverage / counded distribution-mismatch coefficient)**.** We say the environment satisfies coverage if

$$D_\infty \coloneqq \left\| \frac{\mu^\star}{\rho_0} \right\|_\infty = \max_{s\in\mathcal{S}} \frac{\mu^\star(s)}{\rho_0(s)} < \infty, \tag{10}$$

where $\mu^\star = \mu^{\pi^\star}$ and $\pi^\star = \arg\min_{\pi\in\Delta(\mathcal{A})^\mathcal{S}} V(\pi)$.

We note that while it makes sense to consider coverage subject to the best-in-class policy, it is usually considered in conjunction with completeness and therefore in the realizable setting. The following is a notion of approximate completeness that has appeared in e.g., Scherrer and Geist [2014], Bhandari and Russo [2024].

Definition 4 (Approximate completeness / approximate closure)**.** The completeness error of a policy class Π is defined by:

$$\mathcal{E}(\Pi) \coloneqq \max_{\pi\in\Pi} \left\{ \max_{\pi^+\in\Delta(\mathcal{A})^\mathcal{S}} \mathbb{E}_{s\sim\mu^\pi} \left\langle Q^\pi_s, \pi_s - \pi^+_s \right\rangle - \max_{\tilde{\pi}\in\Pi} \mathbb{E}_{s\sim\mu^\pi} \left\langle Q^\pi_s, \pi_s - \tilde{\pi}_s \right\rangle \right\}. \tag{11}$$

We note that as long as we consider strictly stochastic policies with occupancy measures that have full support, $\mathcal{E}(\Pi) = 0$ implies Π is complete. We have the following relation between (approximate) completeness, coverage, and the VGD condition. For proof see Sherman et al. [2025], which in itself is based on [Agarwal et al., 2021, Bhandari and Russo, 2024].

Lemma 1. *Let Π be a policy class and suppose coverage holds, i.e., $D_\infty < \infty$. Then Π satisfies $(\nu, \varepsilon_{\text{vgd}})$-VGD with $\nu = HD_\infty$ and $\varepsilon_{\text{vgd}} = \mathcal{E}(\Pi)H^2 D_\infty$. In particular, if Π is complete it satisfies $(HD_\infty, 0)$-VGD.*

We note that the VGD error floor in the above lemma is identical to the error floor in the convergence guarantees of CPI subject to approximate completeness (Scherrer and Geist, 2014, see also Agarwal et al., 2019). (Recall that $H \coloneqq \frac{1}{1-\gamma}$ denotes the effective horizon.)

Comparison with algorithms of interest. In what follows, we discuss some features of the algorithms listed in Table 1.

- **CPI** [Kakade and Langford, 2002]. As discussed in the introduction and Section 4.2, CPI is not actor-oracle efficient. The bound with the completeness-error error floor does not capture non-realizable setups where best-in-class convergence is possible, subject to the VGD condition (see Sherman et al., 2025 for a simple example).
- **Log-linear NPG** [Agarwal et al., 2021]. In the log-linear NPG setup the policy class is parametrized by softmax-over-linear functions, where linear is w.r.t. given state-action features. When the linear function class can represent all action-value functions with zero-error, the policy class is essentially complete (formally, it is approximately complete with any desired non-zero error.). $\varepsilon_{\text{approx}}$ stands for a bound on the least-squares error of the approximation. The relative condition number κ is defined in Assumption 6.2. Theorem 20 in their work gives a sample complexity upper bound for Q-NPG, which is a version of PMD with linear function approximation suited for the approximately-complete Π learning setup.
- **Log-linear NPG** [Yuan et al., 2022]. $\varepsilon_{\text{approx}}$ is as in the work of Agarwal et al. [2021], C_ν is a concentrability coefficient that is generally stronger than D_∞, and relates to how well are occupancy measures of policies chosen by the algorithm supported on the optimal policy. Their sample complexity for Q-NPG requires an additional relative condition number assumption similar to κ that is not present in the iteration complexity upper bound.
- **PMD** [Alfano and Rebeschini, 2022]. Here, the policy class is composed from a general parametrization of the PMD dual variables, and an exact mirror-and-project function. $\varepsilon_{\text{pmdc}}$ quantifies the least-square error in approximating the dual variables of policies chosen by the exact PMD step. Roughly speaking, this is a generalization of the $\varepsilon_{\text{approx}}$ of Agarwal et al. [2019] to the more general parametrization setup they consider. A sample complexity result is given for the specific case of a shallow neural network parametrization.
- **PMD** [Sherman et al., 2025]. Here, the policy class parametrization is completely general. Sample complexity follows by arguments we present in Appendix E. The analysis of Sherman et al. [2025] applies for non-convex policy classes, but only subject to PMD completeness (i.e., with dependence on $\varepsilon_{\text{pmdc}}$).

A.1 Additional Related work

Policy learning in the tabular or function approximation setups. Prototypical policy optimization methods in the tabular setup include variants of the PMD algorithm [Tomar et al., 2020, Xiao, 2022, Lan, 2023], most commonly negative-entropy regularized PMD which is also known to be equivalent to the Natural Policy Gradient (NPG; Kakade, 2001). Additional works that study PMD in the tabular setup include Geist et al. [2019], Lan [2023], Johnson et al. [2023], Zhan et al. [2023]. The modern analyses of PMD build on the early work of Even-Dar et al. [2009] and online mirror descent [Beck and Teboulle, 2003, Nemirovskij and Yudin, 1983] and as such require some form of completeness of the policy class. An exception is the recent work of Sherman et al. [2025] where PMD is instead cast as a Bregman proximal point method, thus relaxing completeness conditions by relying instead on the VGD assumption.

The majority of recent works into policy optimization with function approximation focus on PMD variants [e.g., Ju and Lan, 2022, Grudzien et al., 2022, Alfano et al., 2023, Yuan et al., 2023] or policy gradients in parameter space [e.g., Zhang et al., 2020, Mei et al., 2020, 2021, Yuan et al., 2022, Mu and Klabjan, 2024]. The influential works of Bhandari and Russo [2024], Agarwal et al. [2021] set the stage for modern research works both into PMD and policy gradients. The notion of variational gradient dominance (or variants thereof) has appeared in several works in the context of policy learning, mostly in relation to policy gradient methods in parameter space or in the tabular setup [Mei et al., 2020, Agarwal et al., 2021, Xiao, 2022, Bhandari and Russo, 2024].

The recent works of Jia et al. [2023], Krishnamurthy et al. [2025] study the boundaries of PAC learnability of policy learning, focusing on forms of environment access, refined policy class conditions, and / or specific structural environment models such as the Block MDP [Krishnamurthy et al., 2016]. Notably, works on agnostic policy learning are comparatively scarce, and mostly focus on specific environment structures [Sekhari et al., 2021]. More generally, there exist a myriad of works that

study RL with function approximation (some of which may be classified as studies of policy learning) subject to particular environment structure, which we briefly review below.

Approximate policy iteration methods. Another class of prototypical policy learning algorithms directly to related our work are approximate policy iteration methods, which in particular include CPI [Kakade and Langford, 2002], as well as, for example, API [Bertsekas and Tsitsiklis, 1996] and PSDP [Bagnell et al., 2003]. Scherrer and Geist [2014] provide performance bounds for CPI subject to the completeness error (see also Agarwal et al., 2019). To our knowledge, all results for approximate policy iteration methods require completeness of the policy class either directly or indirectly (by quantifying the error w.r.t. to completeness). In particular, this is true for the algorithms presented in the work of Scherrer [2014], which provides a thorough comparison of different approximate policy iteration schemes, as well as additional convergence bounds. These include an infinite horizon version of PSDP, and a faster rate for a variant of CPI based on line search and / or subject to stronger concentrability assumptions. A bounded distribution mismatch coefficient D_∞ is the weakest among forms of concentrability [Munos, 2003, 2005, Chen and Jiang, 2019], and it too, as mentioned priorly, is deemed too strong to hold in large scale problems. Notably, the infinite horizon version of PSDP requires non-stationary policies and therefore does not fit into the policy learning model we consider here. In addition, the improved rates obtained for CPI require stronger concentrability assumptions and are therefore in applicable in the setting we consider here.

RL with function approximation more generally. There is a rich literature on RL with function approximation that focuses on setups where the environment exhibits some form of inherent structure [Jiang et al., 2017, Dong et al., 2020, Jin et al., 2020, 2021, Du et al., 2021]. One popular variant for which statistical and computational efficient policy learning is possible is the Linear MDP Yang and Wang [2019, 2020], Jin et al. [2020], and more generally the low-rank MDP [Jiang et al., 2017, Agarwal et al., 2020] where the state-action feature are not given to the learner. Our line of inquiry in this work aims at having no explicit structural assumptions on the environment. We adopt instead an optimization flavored assumption on the relation of the policy class to the landscape of the objective function, which turns out to be weaker than the standard completeness and coverage. As such, our work is better understood as extending the lines of work mentioned in the two preceding paragraphs.

B Additional preliminaries

Discounted MDPs. A discounted Markov Decision Process (MDP; Puterman [1994]) $\mathcal{M}$ is defined by a tuple $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathbb{P}, r, \gamma, \rho_0)$. For notational convenience, for $s, a \in \mathcal{S} \times \mathcal{A}$ we let $\mathbb{P}_{s,a} \coloneqq \mathbb{P}(\cdot \mid s, a) \in \Delta(\mathcal{S})$ denote the next state probability measure. We denote the *value* of π when starting from a state $s \in \mathcal{S}$ by $V_s(\pi)$:

$$V_s(\pi) \coloneqq \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \mid s_0 = s, \pi\right],$$

and more generally for any $\rho \in \Delta(\mathcal{S})$, $V_\rho(\pi) \coloneqq \mathbb{E}_{s\sim\rho} V_s(\pi)$. When the subscript is omitted, $V(\pi)$ denotes value of π when starting from the initial state distribution ρ_0:

$$V(\pi) \coloneqq V_{\rho_0}(\pi) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \mid s_0 \sim \rho_0, \pi\right].$$

For any state action pair $s, a \in \mathcal{S} \times \mathcal{A}$, the action-value function of π, or Q-function, measures the value of π when starting from s, taking action a, and then following π for the reset of the interaction:

$$Q^\pi_{s,a} \coloneqq \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \mid s_0 = s, a_0 = a, \pi\right]$$

We further denote the discounted state-occupancy measure of π induced by any start state distribution $\rho \in \Delta(\mathcal{S})$ by μ^π_ρ:

$$\mu^\pi_\rho(s) \coloneqq (1-\gamma) \sum_{t=0}^{\infty} \gamma^t \Pr(s_t = s \mid s_0 \sim \rho, \pi).$$

It is easily verified that $\mu^\pi \in \Delta(\mathcal{S})$ is indeed a state probability measure. In the sake of brevity, we take the MDP true start state distribution ρ_0 as the default in case one is not specified:

$$\mu^\pi \coloneqq \mu^\pi_{\rho_0}. \tag{12}$$

Problem setup. We consider an infinite horizon environment interaction model (see Protocol 1) similar to that of, e.g., Agarwal et al. [2021].

Protocol 1 Infinite horizon environment rollout

Environment resets $s_0 \sim \rho_0$
for $t = 0, \ldots :$ **do**
  Agent observes s_t, chooses a_t, incurs $r(s_t, a_t)$
  Environment transitions $s_{t+1} \sim \mathbb{P}(\cdot|s_t, a_t)$
  Agent decides whether to terminate episode
end for

Additional notation. For a given set $\mathcal{X}$, we let $\mathcal{N}(\epsilon, \mathcal{X}, \|\cdot\|)$ denote the *ϵ-covering number* of $\mathcal{X}$ w.r.t. the norm $\|\cdot\|$. The convex closure of $\mathcal{X}$ is denoted by $\mathrm{conv}(\mathcal{X})$. We recall our definition for an $\varepsilon_{\mathrm{ex}}$-greedy exploratory version of a policy class Π (see Eq. (7)):

$$\Pi^{\varepsilon_{\mathrm{ex}}} \coloneqq \left\{(1-\varepsilon_{\mathrm{ex}})\pi + \varepsilon_{\mathrm{ex}} u \mid \pi \in \Pi, u_{s,a} \equiv 1/A \ \forall s, a\right\}$$

We conclude by recalling the following notations used throughout:

$$\mu^k \coloneqq \mu^{\pi^k}, \quad Q^k \coloneqq Q^{\pi^k}, \quad S \coloneqq |\mathcal{S}|, \quad A \coloneqq |\mathcal{A}|, \quad H \coloneqq \frac{1}{1-\gamma}.$$

C First Order Methods for non-Euclidean Optimization

In this section, we consider the smooth non-convex optimization problem:

$$\min_{x \in \mathcal{X}} f(x), \tag{13}$$

where $f \colon \mathcal{X} \to \mathbb{R}$ and $\mathcal{X} \subset \mathbb{R}^d$ is a compact convex set. Our reduction detailed in Section 3 leads to the optimization setup described next. All algorithms presented in Section 4 are instances of the algorithms we analyze in this section. Before introducing the problem setup, we give the following definitions.

Definition 5 (Variational Gradient Dominance)**.** We say $f \colon \mathcal{X} \to \mathbb{R}$ satisfies the variational gradient dominance condition with parameters $(\nu, \varepsilon_{\mathrm{vgd}})$, or that f is $(\nu, \varepsilon_{\mathrm{vgd}})$-VGD, if here exist constants $\nu, \varepsilon_{\mathrm{vgd}} > 0$, such that for any $x \in \mathcal{X}$, it holds that:

$$f(x) - \arg\min_{x^\star \in \mathcal{X}} f(x^\star) \leq \nu \max_{\tilde{x} \in \mathcal{X}} \langle \nabla f(x), x - \tilde{x} \rangle + \varepsilon_{\mathrm{vgd}}.$$

Definition 6 (Local Norm)**.** We define a *local* norm over a set $\mathcal{X} \subseteq \mathbb{R}^d$ by a mapping $x \mapsto \|\cdot\|_x$ such that $\|\cdot\|_x$ is a norm for all $x \in \mathcal{X}$. We may denote a local norm by $\|\cdot\|_{(\cdot)}$ or by $x \mapsto \|\cdot\|_x$.

Definition 7 (Local Smoothness)**.** We say $f \colon \mathcal{X} \to \mathbb{R}$ is *β-locally* smooth w.r.t. a local norm $x \mapsto \|\cdot\|_x$ if for all $x, y \in \mathcal{X}$:

$$|f(y) - f(x) - \langle \nabla f(x), y - x \rangle| \leq \frac{\beta}{2} \|y - x\|_x^2 .$$

Definition 8 (Local Lipschitz continuity)**.** We say $f \colon \mathcal{X} \to \mathbb{R}$ is *M-locally* Lipschitz w.r.t. a local norm $x \mapsto \|\cdot\|_x$ if for all $x \in \mathcal{X}$:

$$\|\nabla f(x)\|_x^* \leq M.$$

Problem Setup. We consider the problem of computing an approximate global minimum of the objective Eq. (13), under the following conditions.

Assumption 1. The local norm $x \mapsto \|\cdot\|_x$, decision set $\mathcal{X}$, and objective f satisfy:

- f is β-locally smooth and M-locally Lipschitz w.r.t. $x \mapsto \|\cdot\|_x$ for $M \geq 1$,
- f is $(\nu, \varepsilon_{\text{vgd}})$-VGD over $\mathcal{X}$ with $\nu \geq 1$,
- f is bounded from below; $f^\star = \min_x f(x) > -\infty$,
- $\mathcal{X}$ has a bounded diameter w.r.t. $x \mapsto \|\cdot\|$; $D \geq \max\{1, \max_{x,y,z \in \mathcal{X}} \|z - y\|_x\}$.

The assumption that $D, M, \nu \geq 1$ is solely for simplicity of presentation; if for example the objective f is M-Lipschitz for $M < 1$, our results still hold with $M \to \max\{1, M\}$.

C.1 Constrained steepest descent method

In this section, we consider the constrained steepest descent method; given an initialization $x_1 \in \mathcal{X}$, step size $\eta > 0$, and step error $\varepsilon > 0$:

$$\forall k = 1, \ldots, K,\ x_{k+1} \in \arg\min^{\varepsilon}_{x \in \mathcal{X}} \left\{ \langle \nabla f(x_k), x \rangle + \frac{1}{2\eta} \|x - x_k\|^2_{x_k} \right\}. \tag{14}$$

For this method, we prove the following theorem.

Theorem 5. *Under Assumption 1, the constrained steepest descent method Eq.* (14) *guarantees, as long as $\eta \leq 1/\beta$:*

$$f(x_{K+1}) - f^\star \leq \frac{8\,(\nu M D)^2}{\eta K} + \frac{4\nu M D}{\sqrt{\eta}} \sqrt{\varepsilon} + \varepsilon_{\text{vgd}}$$

To prove Theorem 5, we will first establish an analysis framework that connects the algorithm with a notion of constrained steepest descent magnitude. As a general note, the fact that the norm used in Eq. (14) to compute the steepest descent direction is local makes only a syntactic difference in the analysis. Wherever it is convenient we make a claim about a general norm (which may be local and depend on some point), like in Lemma 2 below.

Let $\eta > 0$ be a fixed step size. Given $x \in \mathcal{X}$, we define the set of potential gradient mappings from x by:

$$\mathcal{G}_x \coloneqq \left\{ \frac{1}{\eta}(x - y) \mid y \in \mathcal{X} \right\}, \tag{15}$$

and the steepest descent magnitude by:

$$\delta_x \coloneqq \max_{g \in \mathcal{G}_x} \left\{ \langle \nabla f(x), g \rangle - \frac{1}{2} \|g\|^2_x \right\}. \tag{16}$$

Lemma 2. *For any norm $\|\cdot\|$, It holds that*

$$x^+ \in \arg\min^{\varepsilon}_{y \in \mathcal{X}} \left\{ \langle \nabla f(x), y \rangle + \frac{1}{2\eta} \|x - y\|^2 \right\}$$
$$\iff \frac{1}{\eta}(x - x^+) \in \arg\max^{(\varepsilon/\eta)}_{g \in \mathcal{G}_x} \left\{ \langle \nabla f(x), g \rangle - \frac{1}{2} \|g\|^2 \right\}.$$

Proof. Observe:

$$\arg\min^{\varepsilon}_{y \in \mathcal{X}} \left\{ \langle \nabla f(x), y \rangle + \frac{1}{2\eta} \|x - y\|^2 \right\} = \arg\min^{\varepsilon}_{y \in \mathcal{X}} \left\{ \langle \nabla f(x), y - x \rangle + \frac{1}{2\eta} \|y - x\|^2 \right\}$$
$$= \arg\min^{(\varepsilon/\eta)}_{y \in \mathcal{X}} \left\{ \left\langle \nabla f(x), \frac{1}{\eta}(y - x) \right\rangle + \frac{1}{2} \left\| \frac{1}{\eta}(y - x) \right\|^2 \right\}.$$

Further, for any $y \in \mathcal{X}$, letting $g = \frac{1}{\eta}(x - y)$, we have

$$\left\langle \nabla f(x), \frac{1}{\eta}(x-y) \right\rangle + \frac{1}{2}\left\| \frac{1}{\eta}(x-y) \right\|^2 = \langle \nabla f(x), -g \rangle + \frac{1}{2}\|g\|^2 .$$

Hence, the gradient mapping $y \mapsto \frac{1}{\eta}(x - y)$ is a bijection between $\mathcal{X}$ and $\mathcal{G}_x$ that gives the same value for the LHS and RHS objectives in the above display. Thus,

$$x^+ \in \arg\min^{\varepsilon}_{y \in \mathcal{X}} \left\{ \langle \nabla f(x), y \rangle + \frac{1}{2\eta}\|x-y\|^2 \right\}$$
$$\iff \frac{1}{\eta}(x - x^+) \in \arg\min^{(\varepsilon/\eta)}_{g \in \mathcal{G}_x} \left\{ \langle \nabla f(x), -g \rangle + \frac{1}{2}\|g\|^2 \right\}.$$

Finally,

$$\arg\min^{\varepsilon}_{g \in \mathcal{G}_x} \left\{ \langle \nabla f(x), -g \rangle + \frac{1}{2}\|g\|^2 \right\} = \arg\max^{(\varepsilon/\eta)}_{g \in \mathcal{G}_x} \left\{ \langle \nabla f(x), g \rangle - \frac{1}{2}\|g\|^2 \right\},$$

which completes the proof. □

Lemma 3. *For all $x \in \mathcal{X}, \eta > 0$, we have:*

$$\max_{y \in \mathcal{X}} \langle \nabla f(x), x - y \rangle \le 2 \max\{D, 1\} \max\left\{ \delta_x, \sqrt{\delta_x} \right\}.$$

Proof. Let $y \in \mathcal{X}$, and note that

$$\langle \nabla f(x), x - y \rangle = \eta \left\langle \nabla f(x), \frac{1}{\eta}(x-y) \right\rangle,$$

hence

$$\max_{y \in \mathcal{X}} \langle \nabla f(x), x - y \rangle = \eta \max_{g \in \mathcal{G}_x} \langle \nabla f(x), g \rangle . \tag{17}$$

In what follows, we consider the set of gradient mappings restricted to direction u and the corresponding descent quantity in direction u:

$$\mathcal{G}_x(u) \coloneqq \mathcal{G}_x \cap \{\alpha u : \alpha \ge 0\}; \quad \delta_x(u) \coloneqq \max_{g \in \mathcal{G}_x(u)} \langle \nabla f(x), g \rangle - \frac{1}{2}\|g\|_x^2 .$$

By Lemma 4, we have:

$$\begin{cases} \delta_x(u) = \frac{1}{2}\langle \nabla f(x), u \rangle^2 & \langle \nabla f(x), u \rangle u \in \mathcal{G}_x, \\ \delta_x(u) \ge \frac{1}{2}\max_{g \in \mathcal{G}_x(u)} \langle \nabla f(x), g \rangle & \langle \nabla f(x), u \rangle u \notin \mathcal{G}_x. \end{cases}$$

Now, since

$$\delta_x = \max_{g \in \mathcal{G}_x} \left\{ \langle \nabla f(x), g \rangle - \frac{1}{2}\|g\|_x^2 \right\} = \max_{\|u\|_x = 1} \delta_x(u),$$

it follows that:

$$\begin{aligned} \max_{g \in \mathcal{G}_x} \langle \nabla f(x), g \rangle &= \max_{u : \|u\|_x = 1} \max_{g \in \mathcal{G}_x(u)} \langle \nabla f(x), g \rangle \\ &\le \max_{u : \|u\|_x = 1} \max\left\{ 2\delta_x(u), \frac{D}{\eta}\sqrt{2\delta_x(u)} \right\} \\ &= \max\left\{ 2\delta_x, \frac{D}{\eta}\sqrt{2\delta_x} \right\} \\ &\le \frac{2}{\eta}\max\{D, 1\} \max\left\{ \delta_x, \sqrt{\delta_x} \right\}. \end{aligned}$$

Combining the above with Eq. (17), the result follows. □

Lemma 4. *Let $x \in \mathcal{X}$, $u \in \mathbb{R}^d$, $\|u\|_x = 1$, and define:*

$$\mathcal{G}_x(u) \coloneqq \mathcal{G}_x \cap \{\alpha u : \alpha \geq 0\}; \quad \delta_x(u) \coloneqq \max_{g \in \mathcal{G}_x(u)} \langle \nabla f(x), g \rangle - \frac{1}{2} \|g\|_x^2 .$$

Then,

$$\begin{cases} \delta_x(u) = \frac{1}{2} \langle \nabla f(x), u \rangle^2 & \langle \nabla f(x), u \rangle u \in \mathcal{G}_x, \\ \delta_x(u) \geq \frac{1}{2} \max_{g \in \mathcal{G}_x(u)} \langle \nabla f(x), g \rangle & \langle \nabla f(x), u \rangle u \notin \mathcal{G}_x. \end{cases}$$

Proof. We have, by definition of $\delta_x(u)$:

$$\delta_x(u) = \max_{\alpha \geq 0 : \alpha u \in \mathcal{G}_x} \left\{ \langle \nabla f(x), \alpha u \rangle - \frac{1}{2} \|\alpha u\|_x^2 \right\},$$

and we note that without constraining $\alpha u \in \mathcal{G}_x$, we have

$$\arg\max_{\alpha \geq 0} \left\{ \langle \nabla f(x), \alpha u \rangle - \frac{1}{2} \|\alpha u\|_x^2 \right\} = \arg\max_{\alpha \geq 0} \left\{ \alpha \langle \nabla f(x), u \rangle - \frac{\alpha^2}{2} \right\} = \langle \nabla f(x), u \rangle .$$

Now, set $A \coloneqq \{\alpha : \alpha u \in \mathcal{G}_x\}$, and $\alpha_0 \coloneqq \sup\{\alpha : \alpha u \in \mathcal{G}_x\}$. Observe that since $\mathcal{G}_x$ is closed and convex, we have $\alpha_0 = \infty \implies A = [0, \infty)$, and $\alpha_0 < \infty \implies A = [0, \alpha_0]$. Proceeding, we now consider the two cases from the lemma statement.

Assume first that $\langle \nabla f(x), u \rangle u \in \mathcal{G}_x$. Then $\alpha_0 \geq \langle \nabla f(x), u \rangle$, and by since $\alpha = \langle \nabla f(x), u \rangle \in A$ minimizes the unconstrained problem, it also minimizes the constrained one, hence

$$\delta_x(u) = \frac{1}{2} \langle \nabla f(x), u \rangle^2 ,$$

as required.

Assume now that $\langle \nabla f(x), u \rangle u \notin \mathcal{G}_x$. Then $\alpha_0 < \langle \nabla f(x), u \rangle$, and since

$$\alpha \mapsto \left\{ \langle \nabla f(x), \alpha u \rangle - \frac{1}{2} \|\alpha u\|_x^2 = \alpha \langle \nabla f(x), u \rangle - \frac{\alpha^2}{2} \right\}$$

is monotonically increasing for $\alpha \in [0, \langle \nabla f(x), \alpha u \rangle]$, we obtain

$$\begin{aligned} \delta_x(u) = \langle \nabla f(x), \alpha_0 u \rangle - \frac{1}{2} \|\alpha_0 u\|_x^2 &= \alpha_0 \left(\langle \nabla f(x), u \rangle - \frac{\alpha_0}{2} \right) \\ &\geq \alpha_0 \left(\langle \nabla f(x), u \rangle - \frac{\langle \nabla f(x), u \rangle}{2} \right) \\ &= \frac{1}{2} \langle \nabla f(x), \alpha_0 u \rangle \\ &= \frac{1}{2} \max_{0 \leq \alpha \in \mathcal{G}_x} \langle \nabla f(x), \alpha u \rangle \\ &= \frac{1}{2} \max_{g \in \mathcal{G}_x(u)} \langle \nabla f(x), g \rangle . \end{aligned}$$

This completes the proof. □

We are now ready for the proof of our main theorem.

Proof of Theorem 5. For ease of presentation, we prove for the case that $\varepsilon_{\text{vgd}} = 0$; the general case follows immediately by replacing $f^\star$ with the error floor $f^\star + \varepsilon_{\text{vgd}}$ everywhere in the proof. Throughout the proof we denote $\mathcal{G}_k \coloneqq \mathcal{G}_{x_k}$ and $\delta_k \coloneqq \delta_{x_k}$. We recall these are the set of potential gradient mappings and steepest descent magnitude Eqs. (15) and (16). For convenience:

$$\mathcal{G}_t = \left\{ \frac{1}{\eta}(x_k - y) \mid y \in \mathcal{X} \right\}; \quad \delta_k = \max_{g \in \mathcal{G}_t} \left\{ \langle \nabla f(x_k), g \rangle - \frac{1}{2} \|g\|_{x_k}^2 \right\}.$$

By Lemma 2, we have that for all k, $g_k \coloneqq \frac{1}{\eta}(x_k - x_{k+1})$ satisfies

$$g_k \in \arg\max^{(\varepsilon/\eta)}_{g \in \mathcal{G}_k} \left\{ \langle \nabla f(x_k), g \rangle - \frac{1}{2} \|g\|^2_{x_k} \right\}.$$

Hence, by smoothness of f:

$$\begin{aligned}
f(x_{k+1}) &\leq f(x_k) + \langle \nabla f(x_k), x_{t=k+1} - x_k \rangle + \frac{\beta}{2} \|x_{k+1} - x_k\|^2_{x_k} \\
&= f(x_k) - \eta \langle \nabla f(x_k), g_k \rangle + \frac{\eta^2 \beta}{2} \|g_k\|^2_{x_k} \\
&\leq f(x_k) - \eta \langle \nabla f(x_k), g_k \rangle + \frac{\eta}{2} \|g_k\|^2_{x_k} && (\eta \leq 1/\beta) \\
&= f(x_k) - \eta \left(\langle \nabla f(x_k), g_k \rangle - \frac{1}{2} \|g_k\|^2_{x_k} \right) \\
&\leq f(x_k) - \eta \left(\delta_k - \varepsilon/\eta \right),
\end{aligned}$$

which implies

$$\eta \delta_k \leq f(x_k) - f(x_{k+1}) + \varepsilon. \tag{18}$$

Further, by the VGD assumption and Lemma 3:

$$\frac{1}{\nu} \left(f(x_k) - f^\star \right) \leq \max_{y \in \mathcal{X}} \langle \nabla f(x_k), x_k - y \rangle \leq 2D \max \left\{ \delta_k, \sqrt{\delta_k} \right\},$$

hence, with $E_k \coloneqq f(x_k) - f^\star$ the above implies

$$\frac{1}{(2D\nu)^2} E_k^2 \leq \max \left\{ \delta_k^2, \delta_k \right\}. \tag{19}$$

In addition, we have

$$\begin{aligned}
\delta_k &= \max_{g \in \mathcal{G}_k} \left\{ \langle \nabla f(x_k), g \rangle - \frac{1}{2} \|g\|^2_{x_k} \right\} \\
&\leq \max_{g \in \mathbb{R}^d} \left\{ \langle \nabla f(x_k), g \rangle - \frac{1}{2} \|g\|^2_{x_k} \right\} = \max_{\|u\|_{x_k} = 1} \langle \nabla f(x_k), u \rangle^2 = \left(\|\nabla f(x_k)\|^*_{x_k} \right)^2 \leq M^2, \\
\implies \max \left\{ \delta_k^2, \delta_k \right\} &\leq \max \left\{ M^2 \delta_k, \delta_k \right\} = M^2 \delta_k. && (M \geq 1)
\end{aligned}$$

Combining with Eq. (19) we obtain

$$\frac{1}{(2D\nu)^2} E_k^2 \leq M^2 \delta_k \implies (2M\nu D)^{-2} E_k^2 \leq \delta_k.$$

Further combining the above display with Eq. (18), we obtain:

$$\begin{aligned}
\omega_0 E_k^2 \leq \eta \delta_k \leq f(x_k) - f(x_{k+1}) + \varepsilon = E_k - E_{k+1} + \varepsilon \\
\iff \omega_0 \left(E_k^2 - \varepsilon/\omega_0 \right) \leq E_k - E_{k+1}.
\end{aligned}$$

for $\omega_0 \coloneqq \eta \left(2\nu M D\right)^{-2}$. We now consider two cases. In the first, the algorithm converges to the error floor determined by ε, in the second, $E_k^2 \geq 2\varepsilon/\omega_0$ for all k.

Case 1 (Convergence to error floor). Suppose that $E_{k_0}^2 \leq 2\varepsilon/\omega_0$ for some $k_0 \in [K]$. By our previous display, it holds that for all k,

$$E_{k+1} \leq E_k - \omega_0 \left(E_k^2 - \varepsilon/\omega_0 \right),$$

which implies that whenever $E_k^2 \geq 2\varepsilon/\omega_0$, $E_{k+1} \leq E_k$. Further, since $E_k \geq 0$, we also have that in any case, $E_{k+1} \leq E_k + \varepsilon$. Now suppose by contradiction that $E_{K+1} > \sqrt{2\varepsilon/\omega_0} + \varepsilon$. This implies that the last iteration was a descent iteration, hence $E_K > \sqrt{2\varepsilon/\omega_0} + \varepsilon$. Proceeding with this argument inductively contradicts our assumption that $E_{k_0}^2 \leq 2\varepsilon/\omega_0$. Thus, we obtain

$$E_{K+1} \leq \sqrt{2\varepsilon/\omega_0} + \varepsilon \leq \frac{4\nu M D}{\sqrt{\eta}} \sqrt{\epsilon}.$$

Case 2 (Descent throughout). Suppose that "Case 1" does not occur, then $E_k^2 \geq 2\varepsilon/\omega_0$ for all k, which implies $\eta\delta_k - \epsilon \geq \eta\delta_k/2$, and for $\omega \coloneqq \omega_0/2$:

$$\omega E_k^2 \leq E_t - E_{t+1}.$$

Now, divide both sides of the previous display by $E_k E_{k+1}$ and use that $E_k \geq E_{k+1}$,

$$\omega \leq \frac{\omega E_k}{E_{k+1}} \leq \frac{1}{E_{k+1}} - \frac{1}{E_k},$$

and sum over k, telescoping the RHS to obtain

$$\begin{aligned} \omega K \leq \frac{1}{E_{K+1}} - \frac{1}{E_1} &\implies \omega K(E_{K+1}E_1) \leq E_1 - E_{K+1} \\ &\implies E_{K+1} \leq E_1 - \omega(E_{K+1}E_1)K. \end{aligned}$$

Finally,

$$0 \leq E_{K+1} \leq E_1\left(1 - \omega E_{K+1}K\right),$$

and dividing by E_1 (if $E_1 = 0$, there is nothing to prove), we obtain

$$0 \leq 1 - \omega E_{K+1}K \implies E_{K+1} \leq \frac{1}{\omega K} = \frac{2\left(2\nu MD\right)^2}{\eta K},$$

and which completes the proof. □

Next, we additionally provide a proof for convergence to an approximate stationary point without the VGD assumption. Here we prove for the error free case

Theorem 6. *Assume that $f\colon \mathcal{X} \to \mathbb{R}$ is β-smooth w.r.t. a norm $\|\cdot\|$ over $\mathcal{X}$ and attains a minimum $f^\star = \min_{x\in\mathcal{X}} f(x)$. Then the constrained steepest descent method Eq.* (14) *with step size $\eta \leq 1/\beta$ guarantees that after $K \geq 1$ iterations, we have that for some $k \in [K]$, x_k is an approximate stationary point:*

$$\min_{y\in\mathcal{X}} \langle \nabla f(x_k), y - x_k\rangle \geq -2D \max\left\{\frac{E_1}{\eta K} + \varepsilon/\eta, \sqrt{\frac{E_1}{\eta K} + \varepsilon/\eta}\right\},$$

where $E_1 \coloneqq f(x_1) - f^\star$.

Proof. Similar to the proof of Theorem 5, we obtain for all k:

$$\eta\delta_k \leq f(x_k) - f(x_{k+1}) + \varepsilon. \tag{20}$$

Summing over k and rearranging,

$$\sum_{t=1}^{T} \delta_k \leq \frac{f(x_1) - f(x_{T+1})}{\eta} + \frac{K\varepsilon}{\eta},$$

which implies that for some k,

$$\delta_k \leq \frac{f(x_1) - f(x_\star)}{\eta K} + \frac{\varepsilon}{\eta}.$$

By Lemma 3, we have

$$\max_{y\in\mathcal{X}} \langle \nabla f(x_k), x_k - y\rangle \leq 2D \max\left\{\delta_k, \sqrt{\delta_k}\right\} \leq 2D \max\left\{\frac{E_1}{\eta K} + \varepsilon/\eta, \sqrt{\frac{E_1}{\eta K} + \varepsilon/\eta}\right\},$$

which proves our claim. □

Algorithm 4 Approximate Frank-Wolfe

input: $\eta_1, \ldots, \eta_K$; error tolerance $\epsilon > 0$.
for $k = 1, \ldots, K$ **do**
  Compute $\tilde{x}_{k+1} \in \arg\min^{\epsilon}_{x \in \mathcal{X}} \langle x, \nabla f(x_k) \rangle$
  Set $x_{k+1} = (1 - \eta_k) x_k + \eta_k \tilde{x}_{k+1}$.
end for

C.2 Frank-Wolfe method

We first present the guarantee of the standard FW [Frank and Wolfe, 1956] method subject to the VGD condition, without local norms and without a second approximation step. The analysis is fairly standard, but uses the VGD condition where convexity is normally used.

Theorem 7. *Let $\|\cdot\|$ be a norm, and $\mathcal{X} \subset \mathbb{R}^d$ be a set of bounded diameter $D \geq \max_{x,y \in \mathcal{X}} \|x - y\|$. Assume that $f : \mathcal{X} \to \mathbb{R}$ is β-smooth w.r.t. $\|\cdot\|$ over $\mathrm{conv}(\mathcal{X})$ and attains a minimum $f^\star = \min_{x \in \mathcal{X}} f(x)$. Assume further that $\mathcal{X}, f$ satisfy a $(\nu, \varepsilon_{\mathrm{vgd}})$-VGD condition. Then the FW method (Algorithm 4) guarantees for all k:*

$$f(x_{k+1}) - f^\star \leq \prod_{s=1}^{k} (1 - \eta_s/\nu) \left(f(x_1) - f^\star \right) + \frac{1}{2} \sum_{s=1}^{k} \prod_{s'=s+1}^{k} (1 - \eta_{s'}/\nu) \left(\eta_s^2 \beta D^2 + 2\eta_s \epsilon \right) + \varepsilon_{\mathrm{vgd}}.$$

Furthermore, with $\eta_k = \frac{2\nu}{k+2}$, we obtain after K iterations:

$$f(x_{K+1}) - f^\star \leq \frac{f(x_1) - f^\star + 2\nu^2 \beta D^2}{K+2} + 2\nu\epsilon + \varepsilon_{\mathrm{vgd}}.$$

Proof. For ease of presentation, we prove for the case that $\varepsilon_{\mathrm{vgd}} = 0$; the general case follows immediately by replacing $f^\star$ with the error floor $f^\star + \varepsilon_{\mathrm{vgd}}$ everywhere in the proof. Observe,

$$\begin{aligned}
f(x_{k+1}) - f(x_k) &\leq \nabla f(x_k)^\top (x_{k+1} - x_k) + \frac{\beta}{2} \|x_{k+1} - x_k\|^2 && (\beta\text{-smoothness}) \\
&= \eta_k \nabla f(x_k)^\top (\tilde{x}_{k+1} - x_k) + \frac{\beta \eta_k^2}{2} \|\tilde{x}_{k+1} - x_k\|^2 && (x_{k+1} - x_k = \eta_k(\tilde{x}_{k+1} - x_k)) \\
&\leq \eta_k \nabla f(x_k)^\top (\tilde{x}_{k+1} - x_k) + \frac{\eta_k^2 \beta D^2}{2}.
\end{aligned}$$

Further by definition of the algorithm,

$$\begin{aligned}
\eta_k \nabla f(x_k)^\top (\tilde{x}_{k+1} - x_k) &\leq \eta_k \min_{\tilde{x} \in \mathcal{X}} \nabla f(x_k)^\top (\tilde{x} - x_k) + \epsilon \eta_k \\
&= \frac{\eta_k}{\nu} \nu \min_{\tilde{x} \in \mathcal{X}} \nabla f(x_k)^\top (\tilde{x} - x_k) + \epsilon \eta_k \\
&\leq \frac{\eta_k}{\nu} \left(f^\star - f(x_k) \right) + \epsilon \eta_k,
\end{aligned}$$

where the last inequality follows by the VGD assumption. Combining this with our previous display now yields,

$$f(x_{k+1}) - f(x_k) \leq \frac{\eta_k}{\nu} \left(f^\star - f(x_k) \right) + \eta_k \epsilon + \beta \eta_k^2 D^2 / 2,$$

hence, letting $E_k \coloneqq f(x_k) - f^\star$ we have,

$$E_{k+1} \leq \left(1 - \frac{\eta_k}{\nu} \right) E_k + \eta_k^2 \beta D^2 / 2 + \eta_k \epsilon.$$

Now, apply the above inequality recursively to obtain

$$\begin{aligned}
E_{k+1} \leq \prod_{s=1}^{k} (1 - \eta_s/\nu) E_1 &+ \frac{1}{2} \sum_{s=1}^{k} \prod_{s'=s+1}^{k} (1 - \eta_{s'}/\nu) \left(\eta_s^2 \beta D^2 \right) \\
&+ \sum_{s=1}^{k} \prod_{s'=s+1}^{k} (1 - \eta_{s'}/\nu) \eta_s \epsilon,
\end{aligned}$$

which proves the first part. For the second part, note that choosing $\eta_t = \frac{2\nu}{t+2}$ gives

$$\prod_{s=1}^{k}(1-\eta_s/\nu) = \prod_{s=1}^{k}\frac{s}{s+2} = \frac{1}{(k+1)(k+2)},$$

and,

$$\prod_{s'=s+1}^{k}(1-\eta_{s'}/\nu)\eta_s = \frac{(s+1)(s+2)}{(k+1)(k+2)}\frac{2\nu}{(s+2)} \leq \frac{2\nu}{k+1}$$

$$\prod_{s'=s+1}^{k}(1-\eta_{s'}/\nu)\eta_s^2 = \frac{(s+1)(s+2)}{(k+1)(k+2)}\frac{4\nu^2}{(s+2)^2} \leq \frac{4\nu^2}{(k+1)(k+2)}.$$

Plugging this back into our bound on $E_{k+1} = f(x_{k+1}) - f^\star$, we obtain:

$$\begin{aligned} f(x_{k+1}) - f(x^\star) &\leq \frac{f(x_1)-f^\star}{(k+1)(k+2)} + \frac{1}{2}\sum_{s=1}^{k}\frac{4\nu^2\beta D^2}{(k+1)(k+2)} + \sum_{s=1}^{k}\frac{2\nu\epsilon}{k+1} \\ &\leq \frac{f(x_1)-f^\star}{(k+1)(k+2)} + \frac{2\nu^2\beta D^2}{k+2} + 2\nu\epsilon \\ &\leq \frac{f(x_1)-f^\star+2\nu^2\beta D^2}{k+2} + 2\nu\epsilon, \end{aligned}$$

as claimed. □

Next, we present the guarantee for the doubly approximate version of FW with local norms.

Algorithm 5 Doubly Approximate Frank-Wolfe

input: $\eta_1, \ldots, \eta_K$; error tolerances $\epsilon, \tilde{\epsilon} > 0$.
for $k = 1, \ldots, K$ **do**
  Compute $\tilde{x}_{k+1} \in \arg\min^{\epsilon}_{x\in\mathcal{X}} \langle x, \nabla f(x_k)\rangle$
  Compute $x_{k+1} \in \arg\min^{\tilde{\epsilon}}_{x\in\mathcal{X}} \left\{\|x - ((1-\eta_k)x_k + \eta_k\tilde{x}_{k+1}))\|^2_{x_k}\right\}$.
end for

Theorem 8. *Under Assumption 1, the Doubly Approximate FW Algorithm 5 guarantees, for all k:*

$$\begin{aligned} f(x_{k+1}) - f^\star \leq &\prod_{s=1}^{k}(1-\eta_s/\nu)\left(f(x_1)-f^\star\right) \\ &+ \frac{1}{2}\sum_{s=1}^{k}\prod_{s'=s+1}^{k}(1-\eta_{s'}/\nu)\left(\eta_s^2\beta D^2 + 2\eta_s\left(\epsilon+\beta\sqrt{\tilde{\epsilon}}\right) + \tilde{\epsilon}(2M+\beta)\right) + \varepsilon_{\text{vgd}}. \end{aligned}$$

Furthermore, with $\eta_k = \frac{2\nu}{k+2}$, we obtain after K iterations:

$$f(x_{K+1}) - f^\star \leq \frac{(2\nu^2+1)\beta D^2}{K+2} + 2\nu\left(\epsilon+\beta\sqrt{\tilde{\epsilon}}\right) + \tilde{\epsilon}\,(M+\beta)\,K + \varepsilon_{\text{vgd}}.$$

Proof. For ease of presentation, we prove for the case that $\varepsilon_{\text{vgd}} = 0$; the general case follows immediately by replacing $f^\star$ with the error floor $f^\star + \varepsilon_{\text{vgd}}$ everywhere in the proof. For all k, let

$$x^\star_{k+1} = (1-\eta_k)x_k + \eta_t\tilde{x}_{k+1}.$$

We have,

$$\begin{aligned} \frac{\beta}{2}\|x_{k+1}-x_k\|^2_{x_k} &= \frac{\beta}{2}\left\|x^\star_{k+1} - x_k + \left(x_{k+1} - x^\star_{k+1}\right)\right\|^2_{x_k} \\ &\leq \frac{\beta}{2}\left\|x^\star_{k+1}-x_k\right\|^2_{x_k} + \beta\left\|x^\star_{k+1}-x_k\right\|\left\|x_{k+1}-x^\star_{k+1}\right\|_{x_k} + \frac{\beta}{2}\left\|x_{k+1}-x^\star_{k+1}\right\|^2_{x_k} \\ &= \frac{\beta\eta_k^2}{2}\|\tilde{x}_{k+1}-x_k\|^2_{x_k} + \beta\eta_k\|\tilde{x}_{k+1}-x_k\|_{x_k}\left\|x_{k+1}-x^\star_{k+1}\right\|_{x_k} + \frac{\beta}{2}\left\|x_{k+1}-x^\star_{k+1}\right\|^2_{x_k} \\ &\leq \frac{\beta\eta_k^2}{2}D^2 + \beta\eta_k\sqrt{\tilde{\epsilon}} + \frac{\beta}{2}\tilde{\epsilon}. \end{aligned}$$

Hence,

$$
\begin{aligned}
f(x_{k+1}) - f(x_k) &\le \nabla f(x_k)^\top (x_{k+1} - x_k) + \frac{\beta}{2} \|x_{k+1} - x_k\|_{x_k}^2 \\
&\le \nabla f(x_k)^\top (x_{k+1}^\star - x_k) + \tilde{\epsilon} M + \frac{\beta}{2} \|x_{k+1} - x_k\|_{x_k}^2 \\
&= \eta_k \nabla f(x_k)^\top (\tilde{x}_{k+1} - x_k) + \tilde{\epsilon} M + \frac{\beta}{2} \|x_{k+1} - x_k\|_{x_k}^2 \\
&\le \eta_k \nabla f(x_k)^\top (\tilde{x}_{k+1} - x_k) + \tilde{\epsilon} M + \beta \eta_k^2 D^2/2 + \beta \eta_k \sqrt{\tilde{\epsilon}} + \beta \tilde{\epsilon}/2
\end{aligned}
$$

Now, note that by definition of the algorithm,

$$
\begin{aligned}
\eta_k \nabla f(x_k)^\top (\tilde{x}_{k+1} - x_k) &\le \eta_k \min_{\tilde{x} \in \mathcal{X}} \nabla f(x_k)^\top (\tilde{x} - x_k) + \epsilon \eta_k \\
&= \frac{\eta_k}{\nu} \nu \min_{\tilde{x} \in \mathcal{X}} \nabla f(x_k)^\top (\tilde{x} - x_k) + \epsilon \eta_k \\
&\le \frac{\eta_k}{\nu} \left(f^\star - f(x_k) \right) + \epsilon \eta_k,
\end{aligned}
$$

where the last inequality follows by the VGD assumption. Combining this with our previous display now yields,

$$
\begin{aligned}
f(x_{k+1}) - f(x_k) &\le \frac{\eta_k}{\nu} \left(f^\star - f(x_k) \right) + \eta_k \epsilon + \tilde{\epsilon} M + \beta \eta_k^2 D^2/2 + \beta \eta_k \sqrt{\tilde{\epsilon}} + \beta \tilde{\epsilon}/2 \\
&= \frac{\eta_k}{\nu} \left(f^\star - f(x_k) \right) + \eta_k^2 \beta D^2/2 + \eta_k \left(\epsilon + \beta \sqrt{\tilde{\epsilon}} \right) + \tilde{\epsilon} \left(M + \beta/2 \right),
\end{aligned}
$$

hence, letting $E_k \coloneqq f(x_k) - f^\star$ we have,

$$
E_{k+1} \le \left(1 - \frac{\eta_k}{\nu} \right) E_k + \eta_k^2 \beta D^2/2 + \eta_k \left(\epsilon + \beta \sqrt{\tilde{\epsilon}} \right) + \tilde{\epsilon} \left(M + \beta/2 \right).
$$

Now, apply the above inequality recursively to obtain

$$
\begin{aligned}
E_{k+1} \le \prod_{s=1}^{k} (1 - \eta_s/\nu) E_1 &+ \frac{1}{2} \sum_{s=1}^{k} \prod_{s'=s+1}^{k} (1 - \eta_{s'}/\nu) \left(\eta_s^2 \beta D^2 \right) \\
&+ \sum_{s=1}^{k} \prod_{s'=s+1}^{k} (1 - \eta_{s'}/\nu) \eta_s \left(\epsilon + \beta \sqrt{\tilde{\epsilon}} \right) \\
&+ \sum_{s=1}^{k} \prod_{s'=s+1}^{k} (1 - \eta_{s'}/\nu) \tilde{\epsilon} \left(M + \beta/2 \right),
\end{aligned}
$$

which proves the first part. For the second part, note that choosing $\eta_t = \frac{2\nu}{t+2}$ gives

$$
\prod_{s=1}^{k} (1 - \eta_s/\nu) = \prod_{s=1}^{k} \frac{s}{s+2} = \frac{1}{(k+1)(k+2)},
$$

and,

$$
\begin{aligned}
\prod_{s'=s+1}^{k} (1 - \eta_{s'}/\nu) &= \frac{(s+1)(s+2)}{(k+1)(k+2)}, \\
\prod_{s'=s+1}^{k} (1 - \eta_{s'}/\nu) \eta_s &= \frac{(s+1)(s+2)}{(k+1)(k+2)} \frac{2\nu}{(s+2)} \le \frac{2\nu}{k+1} \\
\prod_{s'=s+1}^{k} (1 - \eta_{s'}/\nu) \eta_s^2 &= \frac{(s+1)(s+2)}{(k+1)(k+2)} \frac{4\nu^2}{(s+2)^2} \le \frac{4\nu^2}{(k+1)(k+2)}.
\end{aligned}
$$

Plugging this back into our bound on $E_{k+1} = f(x_{k+1}) - f^\star$, we obtain, for any $x^\star \in \arg\min_{x\in\mathcal{X}} f(x)$,

$$\begin{aligned}
f(x_{k+1}) - f(x^\star) &\le \frac{f(x_1) - f(x^\star)}{(k+1)(k+2)} + \frac{1}{2}\sum_{s=1}^{k} \frac{4\nu^2\beta D^2}{(k+1)(k+2)} + \sum_{s=1}^{k} \frac{2\nu\left(\epsilon + \beta\sqrt{\tilde{\epsilon}}\right)}{k+1} + \tilde{\epsilon}\,(M+\beta)\,k \\
&\le \frac{f(x_1) - f(x^\star)}{(k+1)(k+2)} + \frac{2\nu^2\beta D^2}{k+2} + 2\nu\left(\epsilon + \beta\sqrt{\tilde{\epsilon}}\right) + \tilde{\epsilon}\,(M+\beta)\,k \\
&\le \frac{\beta\,\|x_1 - x^\star\|_{x^\star}^2}{2(k+1)(k+2)} + \frac{2\nu^2\beta D^2}{k+2} + 2\nu\left(\epsilon + \beta\sqrt{\tilde{\epsilon}}\right) + \tilde{\epsilon}\,(M+\beta)\,k \\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad (\langle \nabla f(x^\star), x_1 - x_\star\rangle \ge 0) \\
&\le \frac{(2\nu^2+1)\beta D^2}{k+2} + 2\nu\left(\epsilon + \beta\sqrt{\tilde{\epsilon}}\right) + \tilde{\epsilon}\,(M+\beta)\,k,
\end{aligned}$$

as claimed. □

C.3 Bregman proximal point method

We first recall the algorithm as presented in Sherman et al. [2025]. Given any convex regularizer $h\colon \mathbb{R}^d \to \mathbb{R}$, we define the set of ϵ-approximate Bregman proximal point update solutions with step-size $\eta > 0$ by:

$$\mathcal{T}_\eta^\epsilon(x; h) := \left\{ x^+ \in \mathcal{X} \mid \forall z \in \mathcal{X} : \left\langle \nabla f(x) + \frac{1}{\eta}\nabla B_h(x^+, x), z - x^+ \right\rangle \ge -\epsilon \right\}. \tag{21}$$

The approximate Bregman proximal point update of Sherman et al. [2025] is defined by:

$$k = 1, \ldots, K: \quad x_{k+1} \in \mathcal{T}_\eta^\epsilon(x_k; R_{x_k}). \tag{22}$$

Theorem 9 (Sherman et al. [2025])**.** *Consider Assumption 1, and suppose further that the local regularizer R_x is 1-strongly convex and has an L-Lipschitz gradient w.r.t. $\|\cdot\|_x$ for all $x \in \mathcal{X}$. Then, for the Bregman proximal point update Eq.* (22) *we have following guarantee, for $\eta \le 1/(2\beta)$:*

$$f(x_{K+1}) - f^\star = O\left(\frac{\nu^2 L^2 c_1^2}{\eta K} + \nu\epsilon + c_1 L \eta^{-\frac{1}{2}}\sqrt{\epsilon} + \varepsilon_{\text{vgd}}\right)$$

where $c_1 := D + \eta M$.

We now consider the Bregman proximal point algorithm that operates with iterates that are approximate minimizers in terms of *function values*:

$$k = 1, \ldots, K: \quad x_{k+1} \leftarrow \arg\min_{x\in\mathcal{X}}^{\varepsilon} \left\{ \langle \nabla f(x_k), x\rangle + \frac{1}{\eta} B_{R_{x_k}}(x, x_k) \right\} \tag{23}$$

We will use the following lemma to translate objective sub-optimality to approximate optimality conditions.

Lemma 5. *Let $\|\cdot\|$ be a norm, and suppose $D \ge \max\left\{1, \max_{x,y\in\mathcal{X}} \|x - y\|\right\}$. Let $\phi\colon \mathcal{X} \to \mathbb{R}$ be 1-strongly convex and L-smooth w.r.t.$\|\cdot\|$. Then, for any $\epsilon \le 1$ if $\hat{x} \in \arg\min_{x\in\mathcal{X}}^{\epsilon} \phi(x)$, we have:*

$$\langle \phi(\hat{x}), y - \hat{x}\rangle \ge -2LD\sqrt{2\epsilon}.$$

Proof. Let $x^\star = \arg\min_{x\in\mathcal{X}} \phi(x)$. By 1-strong convexity and our assumption,

$$\frac{1}{2}\left\|\hat{x} - x^\star\right\|^2 \le \phi(\hat{x}) - \phi(x^\star) \le \epsilon \implies \left\|\hat{x} - x^\star\right\| \le \sqrt{2\epsilon}.$$

Hence, for any $y \in \mathcal{X}$, by L-smoothness:

$$\begin{aligned}
\langle \phi(\hat{x}), \hat{x} - y\rangle &= \left\langle \phi(x^\star), \hat{x} - y\right\rangle + \left\langle \phi(\hat{x}) - \phi(x^\star), \hat{x} - y\right\rangle \\
&\le \left\langle \phi(x^\star), \hat{x} - y\right\rangle + LD\sqrt{2\epsilon}
\end{aligned}$$

Further by our assumption and optimality conditions at $x^\star$,

$$\begin{aligned}\left\langle \phi(x^\star), \hat{x} - y \right\rangle &= \left\langle \phi(x^\star), x^\star - y \right\rangle + \left\langle \phi(x^\star), \hat{x} - x^\star \right\rangle \\ &\leq \left\langle \phi(x^\star), x^\star - y \right\rangle + \phi(\hat{x}) - \phi(x^\star) \\ &\leq \left\langle \phi(x^\star), x^\star - y \right\rangle + \epsilon \\ &\leq +\epsilon.\end{aligned}$$

Hence, we have for all $y \in \mathcal{X}$:

$$\langle \phi(\hat{x}), \hat{x} - y \rangle \leq LD\sqrt{2\epsilon} + \epsilon \leq 2LD\sqrt{2\epsilon},$$

which completes the proof. □

We are now in position to tprove the following.

Theorem 10. *In the same setting of Theorem 9, we have that the Bregman proximal method Eq.* (23) *with $\epsilon \leq 1$ guarantees, for $\eta \leq 1/(2\beta)$:*

$$f(x_{K+1}) - f^\star = O\left(\frac{\nu^2 L^2 c_1^2}{\eta K} + \nu LD\sqrt{\epsilon} + \frac{c_1\sqrt{L^3 D}}{\sqrt{\eta}}\epsilon^{1/4} + \varepsilon_{\text{vgd}}\right)$$

where $c_1 \coloneqq D + \eta M$.

Proof. By Lemma 5 applied with the norm $\|\cdot\|_{x_k}$ and $\phi(x) = \langle \nabla f(x_k), x\rangle + \frac{1}{\eta} B_{R_{x_k}}(x, x_k)$, we have that x_{k+1} from Eq. (23) satisfies:

$$\forall z \in \mathcal{X} : \left\langle \nabla f(x_k) + \frac{1}{\eta}\nabla B_{R_{x_k}}(x_{k+1}, x_k), z - x_{k+1} \right\rangle \geq -2LD\sqrt{2\epsilon}.$$

This implies that $x_{k+1} \in \mathcal{T}_\eta^{\tilde{\epsilon}}(x_k; R_{x_k})$ for $\tilde{\epsilon} = 2LD\sqrt{2\epsilon}$. This proves the claimed result be substituting $\epsilon \to 2LD\sqrt{2\epsilon}$ in the bound of Theorem 9. □

D Proofs for Section 4

In this section, we provide the proofs of convergence for our algorithms in their idealized versions. All proofs build on casting our algorithms as instances of those presented in the pure optimization setup, in Appendix C. We note that we did not make particular effort in optimizing dependence on problem parameters other than K, and in some cases intentionally opted for slightly worse dependence in favor of cleaner bounds.

D.1 Analysis preliminaries

Given a state probability measure $\mu \in \Delta(\mathcal{S})$, and an action space norm $\|\cdot\|_\circ : \mathbb{R}^A \to \mathbb{R}$, we define the induced state-action weighted L^p norm $\|\cdot\|_{L^p(\mu),\circ} : \mathbb{R}^{SA} \to \mathbb{R}$:

$$\|u\|_{L^p(\mu),\circ} \coloneqq \left(\mathbb{E}_{s\sim\mu}\|u_s\|_\circ^p\right)^{1/p}. \tag{24}$$

For any norm $\|\cdot\|$, we let $\|\cdot\|^*$ denote its dual. When discussing a generic norm and there is no risk of confusion, we may use $\|\cdot\|_*$ to refer to its dual. In addition, for $\mu \in \mathbb{R}^S, Q \in \mathbb{R}^{SA}$, we define the state to state-action element-wise product $\mu \circ Q \in \mathbb{R}^{SA}$:

$$(\mu \circ Q)_{s,a} \coloneqq \mu(s) Q_{s,a}. \tag{25}$$

Below, we collect a number of results that will be used repeatedly in the analyses.

Lemma 6 (Value difference; Kakade and Langford, 2002)**.** *For any $\rho \in \Delta(\mathcal{S})$,*

$$V_\rho(\tilde{\pi}) - V_\rho(\pi) = \frac{1}{1-\gamma}\mathbb{E}_{s\sim\mu_\rho^\pi}\left\langle Q_s^{\tilde{\pi}}, \tilde{\pi}_s - \pi_s \right\rangle.$$

Lemma 7 (Policy gradient theorem; Sutton et al., 1999)**.** *For any $\rho \in \Delta(\mathcal{S})$,*

$$\left(\nabla V_\rho(\pi)\right)_{s,a} = \frac{1}{1-\gamma}\mu_\rho^\pi(s)Q_{s,a}^\pi,$$
$$\left\langle \nabla V_\rho(\pi), \tilde{\pi} - \pi \right\rangle = \frac{1}{1-\gamma}\mathbb{E}_{s\sim\mu_\rho^\pi}\left\langle Q_s^\pi, \tilde{\pi}_s - \pi_s \right\rangle.$$

Lemma 8 (Sherman et al., 2025)**.** *Let $\pi\colon \mathcal{S} \to \Delta(\mathcal{A})$ be any policy such that $\varepsilon_{\text{ex}} \coloneqq \min_{s,a}\{\pi_{sa}\} > 0$. Then, for any $\tilde{\pi} \in \mathcal{S} \to \Delta(\mathcal{A})$, we have:*

$$|V(\tilde{\pi}) - V(\pi) - \langle \nabla V(\pi), \tilde{\pi} - \pi\rangle| \le \min\left\{\frac{H^3}{\sqrt{\epsilon}}\|\tilde{\pi} - \pi\|_{L^2(\mu^\pi),1}^2, \frac{AH^3}{\sqrt{\epsilon}}\|\tilde{\pi} - \pi\|_{L^2(\mu^\pi),2}^2\right\}.$$

Lemma 9 (Sherman et al., 2025)**.** *Assume Π is $(\nu, \varepsilon_{\text{vgd}})$-VGD w.r.t. $\mathcal{M}$, and consider the ε_{ex}-greedy exploratory version of Π, $\Pi^{\varepsilon_{\text{ex}}} \coloneqq \{(1-\varepsilon_{\text{ex}})\pi + \varepsilon_{\text{ex}}u \mid \pi \in \Pi\}$, where $u_{s,a} \equiv 1/A$. Then $\Pi^{\varepsilon_{\text{ex}}}$ is $(\nu, \tilde{\varepsilon}_{\text{vgd}})$-VGD with $\tilde{\varepsilon}_{\text{vgd}} \coloneqq \varepsilon_{\text{vgd}} + 12\nu H^2 A\varepsilon_{\text{ex}}$.*

The next lemmas follow from standard arguments, for proofs refer to Sherman et al. [2025].

Lemma 10. *For any strictly positive measure $\mu \in \mathbb{R}_{++}^S$, the dual norm of $\|\cdot\|_{L^2(\mu),\circ}$ is given by*

$$\|z\|_{L^2(\mu),\circ}^* = \sqrt{\int \mu(s)^{-1}\left(\|z_s\|_\circ^*\right)^2 \mathrm{d}s} \tag{26}$$

Lemma 11. *Let $\mu \in \Delta(\mathcal{S})$, and consider the state-action norm $\|\cdot\|_{L^2(\mu),\circ}$. For any $W \in \mathbb{R}^{SA}$, we have*

$$\|\mu \circ W\|_{L^2(\mu),\circ}^* = \sqrt{\mathbb{E}_{s\sim\mu}\left(\|W_s\|_\circ^*\right)^2}$$

Lemma 12. *For any policy $\pi \in \Delta(\mathcal{A})^{\mathcal{S}}$ and action norm $\|\cdot\|_\circ : \mathbb{R}^{\mathcal{A}} \to \mathbb{R}$, if $\max_{s\in\mathcal{S}}\left\|Q_s^\pi\right\|_\circ^* \le G$, then it holds that: $\|\nabla V(\pi)\|_{L^2(\mu^\pi),\circ}^* \le HG$.*

Proof. Observe, by Lemma 11:

$$\|\nabla V(\pi)\|_{L^2(\mu^\pi),\circ}^* = H\|\mu^\pi \circ Q^\pi\|_{L^2(\mu^\pi),\circ}^* = H\sqrt{\mathbb{E}_{s\sim\mu^\pi}\left(\|Q_s^\pi\|_\circ^*\right)^2} \le HG.$$

□

D.2 SDPO

Theorem (Restatement of Theorem 1). Let Π be a convex policy class that satisfies $(\nu, \varepsilon_{\text{vgd}})$-VGD w.r.t. $\mathcal{M}$. Suppose that SDPO (Algorithm 1) is executed with the L^1 action norm $\|\cdot\|_1$. Then, after K iterations, with appropriately tuned η and ε_{ex}, the output of SDPO satisfies:

$$V(\hat{\pi}) - V^\star(\Pi) = O\left(\frac{\nu^2 AH^4}{K^{2/3}} + \nu H^3\sqrt{A}K^{1/6}\sqrt{\varepsilon} + \varepsilon_{\text{vgd}}\right).$$

Proof of Theorem 1. By the policy gradient theorem (Lemma 7), the update step in Algorithm 1 may be equivalently written as:

$$\pi^{k+1} \in \arg\min_{\pi\in\Pi^{\varepsilon_{\text{ex}}}}\left\langle \nabla V(\pi^k), \pi\right\rangle + \frac{1}{2\eta}\left\|\pi - \pi^k\right\|_{L^2(\mu^k),1}^2 \tag{27}$$

We now verify a number of conditions that place us in the setup of Assumption 1.

- **Local smoothness.** By Lemma 8 and the definition of $\Pi^{\varepsilon_{\text{ex}}}$, the value function is $(2\sqrt{A}H^3/\sqrt{\varepsilon_{\text{ex}}})$ locally smooth w.r.t. the local norm $\pi \mapsto \|\cdot\|_{L^2(\mu^\pi),1}$.
- **VGD condition for $\Pi^{\varepsilon_{\text{ex}}}$.** By Lemma 9, we have that $\Pi^{\varepsilon_{\text{ex}}}$ satisfies $(\nu, \tilde{\varepsilon}_{\text{vgd}})$ with $\tilde{\varepsilon}_{\text{vgd}} \coloneqq \varepsilon_{\text{vgd}} + 12\nu H^2 A\varepsilon_{\text{ex}}$.

- **Local Lipschitz property.** By Lemma 12, the value function is H^2-local Lipschitz w.r.t. the local norm $\pi \mapsto \|\cdot\|_{L^2(\mu^\pi),1}$.
- **Diameter bound.** We have that $\|\pi' - \tilde{\pi}\|_{L^2(\mu^\pi),1} \leq \max_{p,q\in\Delta(\mathcal{A})} \|p-q\|_1 \leq 2$, for all $\pi, \pi', \tilde{\pi} \in \Delta(\mathcal{A})^{\mathcal{S}}$.

The above imply we are in the setting of Theorem 5 with $\beta = 2\sqrt{A}H^3/\sqrt{\varepsilon_{\text{ex}}}$, $M = H^2, D = 2$. Thus, setting $\eta = \sqrt{\varepsilon_{\text{ex}}}/(2H^3\sqrt{A})$, ensures that after K iterations of Eq. (27), it is guaranteed that:

$$\begin{aligned} V(\pi^{K+1}) - V^\star(\Pi) &\lesssim \frac{\left(\nu H^2\right)^2}{\eta K} + \frac{\nu H^2}{\sqrt{\eta}}\sqrt{\varepsilon} + \tilde{\varepsilon}_{\text{vgd}} \\ &\lesssim \frac{\nu^2\sqrt{A}H^7}{\sqrt{\varepsilon_{\text{ex}}}K} + \frac{\nu H^5 A^{1/4}}{{\varepsilon_{\text{ex}}}^{1/4}}\sqrt{\varepsilon} + \nu A H^2 \varepsilon_{\text{ex}} + \varepsilon_{\text{vgd}}, \end{aligned}$$

where $\lesssim$ hides only universal constant factors. Now set $\varepsilon_{\text{ex}} = \frac{H^2}{K^{2/3}}$, then

$$\begin{aligned} V(\pi^{K+1}) - V^\star(\Pi) &\lesssim \frac{\nu^2 A H^6}{K^{2/3}} + \frac{\nu H^5 A^{1/4}}{{\varepsilon_{\text{ex}}}^{1/4}}\sqrt{\varepsilon} + \varepsilon_{\text{vgd}} \\ &\leq \frac{\nu^2 A H^6}{K^{2/3}} + \nu H^5\sqrt{A}K^{1/6}\sqrt{\varepsilon} + \varepsilon_{\text{vgd}}, \end{aligned}$$

as required. □

D.3 CPI and DA-CPI

In this section, we provide the analysis for CPI and DA-CPI. For CPI, we can make an argument using a non-local norm owed to the usage of the exact convex combination policy obtained in the second step if each iteration in Algorithm 2. Our first lemma below establishes smoothnes of the value function w.r.t. the global $\|\cdot\|_{\infty,1}$ norm.

Lemma 13. *The value function is $(2H^3)$-smooth w.r.t. the $\|\cdot\|_{\infty,1}$ norm; for any $\pi, \tilde{\pi} \in \mathcal{S} \to \Delta(\mathcal{A})$, we have:*

$$|V(\tilde{\pi}) - V(\pi) - \langle \nabla V(\pi), \tilde{\pi} - \pi\rangle| \leq \frac{2H^3}{2}\|\tilde{\pi} - \pi\|_{\infty,1}^2 .$$

Proof. We have by value difference Lemma 6 and the policy gradient theorem Lemma 7:

$$\begin{aligned} |V(\tilde{\pi}) - V(\pi) - \langle \nabla V(\pi), \tilde{\pi} - \pi\rangle| &= \left|H\mathbb{E}_{s\sim\mu^\pi}\left\langle Q_s^{\tilde{\pi}}, \tilde{\pi}_s - \pi_s\right\rangle - H\mathbb{E}_{s\sim\mu^\pi}\left\langle Q_s^{\pi}, \tilde{\pi}_s - \pi_s\right\rangle\right| \\ &= \left|H\mathbb{E}_{s\sim\mu^\pi}\left\langle Q_s^{\tilde{\pi}} - Q_s^{\pi}, \tilde{\pi}_s - \pi_s\right\rangle\right| \\ &\leq H\mathbb{E}_{s\sim\mu^\pi}\left[\left|\left\langle Q_s^{\tilde{\pi}} - Q_s^{\pi}, \tilde{\pi}_s - \pi_s\right\rangle\right|\right] \\ &\leq H\mathbb{E}_{s\sim\mu^\pi}\left[\left\|Q_s^{\tilde{\pi}} - Q_s^{\pi}\right\|_\infty \|\tilde{\pi}_s - \pi_s\|_1\right] . \end{aligned} \tag{28}$$

Further, again by value difference Lemma 6, for any $s, a \in \mathcal{S}\times\mathcal{A}$:

$$\begin{aligned} Q_{s,a}^{\tilde{\pi}} - Q_{s,a}^{\pi} &= \gamma\mathbb{E}_{s'\sim\mathbb{P}_{s,a}}\left[V_{s'}(\tilde{\pi}) - V_{s'}(\pi)\right] \\ &= \gamma H\mathbb{E}_{s'\sim\mathbb{P}_{s,a}}\left[\mathbb{E}_{s''\sim\mu_{s'}^\pi}\left\langle Q_{s''}^{\tilde{\pi}}, \tilde{\pi}_{s''} - \pi_{s''}\right\rangle\right] \\ &= \gamma H\sum_{s'}\mathbb{P}(s'|s,a)\sum_{s''}\mu_{s'}^\pi(s'')\left\langle Q_{s''}^{\tilde{\pi}}, \tilde{\pi}_{s''} - \pi_{s''}\right\rangle \\ &= \gamma H\sum_{s''}\sum_{s'}\mathbb{P}(s'|s,a)\mu_{s'}^\pi(s'')\left\langle Q_{s''}^{\tilde{\pi}}, \tilde{\pi}_{s''} - \pi_{s''}\right\rangle \\ &= \gamma H\sum_{s''}\mu_{\mathbb{P}_{s,a}}^\pi(s'')\left\langle Q_{s''}^{\tilde{\pi}}, \tilde{\pi}_{s''} - \pi_{s''}\right\rangle \\ &= \gamma H\mathbb{E}_{s''\sim\mu_{\mathbb{P}_{s,a}}^\pi}\left\langle Q_{s''}^{\tilde{\pi}}, \tilde{\pi}_{s''} - \pi_{s''}\right\rangle . \end{aligned}$$

This implies that for any s,

$$\begin{aligned}\left\|Q_s^{\tilde{\pi}} - Q_s^{\pi}\right\|_\infty &= \gamma H \max_a \left|\mathbb{E}_{s'\sim\mu^{\pi}_{\mathbb{P}_{s,a}}} \left\langle Q_{s'}^{\tilde{\pi}}, \tilde{\pi}_{s'} - \pi_{s'}\right\rangle\right| \\ &\leq \gamma H^2 \max_a \mathbb{E}_{s'\sim\mu^{\pi}_{\mathbb{P}_{s,a}}} \left\|\tilde{\pi}_{s'} - \pi_{s'}\right\|_1 \\ &\leq \gamma H^2 \left\|\tilde{\pi} - \pi\right\|_{\infty,1}\end{aligned}$$

Plugging the above back into Eq. (28), we obtain

$$\begin{aligned}\left|V^{\tilde{\pi}} - V^{\pi} - \langle \nabla V^{\pi}, \tilde{\pi} - \pi\rangle\right| &\leq \gamma H^3 \left\|\tilde{\pi} - \pi\right\|_{\infty,1} \mathbb{E}_{s\sim\mu^{\pi}} \left\|\tilde{\pi}_s - \pi_s\right\|_1 \\ &\leq \gamma H^3 \left\|\tilde{\pi} - \pi\right\|_{\infty,1}^2,\end{aligned}$$

which completes the proof up to a trivial computation. □

We are now ready to prove the guarantee for CPI (Algorithm 2), by means of reducing it to a non-Euclidean instance of FW Algorithm 4.

Theorem (Restatement of Theorem 2). Let Π be a policy class that satisfies $(\nu, \varepsilon_{\text{vgd}})$-VGD w.r.t. $\mathcal{M}$. Suppose that CPI (Algorithm 2) is executed with the step size choices $\eta_k = \frac{2\nu}{k+2}$ for $k = 1, \ldots, K$. Then, we have the guarantee that:

$$V(\pi^K) - V^\star(\Pi) \leq \frac{8(2\nu^2+1)H^3}{K} + 2\nu\varepsilon + \varepsilon_{\text{vgd}}$$

Proof of Theorem 2. By the policy gradient theorem (Lemma 7), the update step in CPI (Algorithm 2) may be equivalently written as:

$$\pi^{k+1} \in \arg\min^{\varepsilon}_{\pi\in\Pi} \left\langle \nabla V(\pi^k), \pi - \pi^k\right\rangle.$$

We now verify conditions that place us in the setting of Theorem 7.

- **Global smoothness.** By Lemma 13, the value function is $(2H^3)$-smooth w.r.t. the $\|\cdot\|_{\infty,1}$ norm.
- **VGD condition.** By assumption, Π satisfies $(\nu, \varepsilon_{\text{vgd}})$ -VGD.
- **Diameter bound.** We have that $\|\pi - \tilde{\pi}\|_{\infty,1} \leq \max_{p,q\in\Delta(\mathcal{A})} \|p - q\|_1 \leq 2$ for all $\pi, \tilde{\pi} \in \Delta(\mathcal{A})^{\mathcal{S}}$.

The above imply we are in the setting of Theorem 7 with $\beta = 2H^3$ and $D = 2$. Thus, our step size choice ensures that after K iterations, it is guaranteed that:

$$V(\pi^{K+1}) - V^\star(\Pi) \leq \frac{H + 16\nu^2 H^3}{K} + 2\nu\varepsilon + \varepsilon_{\text{vgd}} \leq \frac{(16\nu^2+1)H^3}{K} + 2\nu\varepsilon + \varepsilon_{\text{vgd}},$$

as required. □

Next, we provide the proof for the guarantees of DA-CPI, which relies on the use of local norms and is actor-oracle efficient.

Theorem (Restatement of Theorem 3). Let Π be a convex policy class that satisfies $(\nu, \varepsilon_{\text{vgd}})$-VGD w.r.t. $\mathcal{M}$. Suppose that DA-CPI (Algorithm 3) is executed with the L^1 action norm $\|\cdot\|_1$. Then, for an appropriate setting of $\eta_1, \ldots, \eta_K$ and ε_{ex}, we have that the DA-CPI output satisfies:

$$V(\hat{\pi}) - V^\star(\Pi) = O\left(\nu^2 A H^3 \left(\frac{1}{K^{2/3}} + \varepsilon^{1/3} + \varepsilon^{2/3} K^{2/3}\right)\right).$$

Proof of Theorem 3. By the policy gradient theorem (Lemma 7), the first update step in Algorithm 3 may be equivalently written as:

$$\pi^{k+1} \in \arg\min^{\varepsilon}_{\pi\in\Pi^{\varepsilon_{\text{ex}}}} \left\langle \nabla V(\pi^k), \pi - \pi^k\right\rangle.$$

We now verify a number of conditions that place us in the setup of Assumption 1.

Algorithm 6 Policy Mirror Descent (PMD)

Input: $K \geq 1, \eta > 0, \varepsilon > 0, \varepsilon_{\text{ex}} > 0, \Pi \in \Delta(\mathcal{A})^{\mathcal{S}}$, and action regularizer $R\colon \mathbb{R}^{\mathcal{A}} \to \mathbb{R}$.
Initialize $\pi^1 \in \Pi^{\varepsilon_{\text{ex}}}$
for $k = 1$ **to** K **do**
  Update $\pi^{k+1} \leftarrow \arg\min^{\varepsilon}_{\pi \in \Pi^{\varepsilon_{\text{ex}}}} \mathbb{E}_{s\sim\mu^k}\left[H\left\langle Q^k_s, \pi_s\right\rangle + \frac{1}{\eta} B_R\left(\pi_s, \pi^k_s\right)\right]$
end for
return $\hat{\pi} \coloneqq \pi^{K+1}$

- **Local smoothness.** By Lemma 8 and the definition of $\Pi^{\varepsilon_{\text{ex}}}$, the value function is $(2\sqrt{A}H^3/\sqrt{\varepsilon_{\text{ex}}})$ locally smooth w.r.t. the local norm $\pi \mapsto \|\cdot\|_{L^2(\mu^\pi),1}$.
- **VGD condition for $\Pi^{\varepsilon_{\text{ex}}}$.** By Lemma 9, we have that $\Pi^{\varepsilon_{\text{ex}}}$ satisfies $(\nu, \tilde{\varepsilon}_{\text{vgd}})$ with $\tilde{\varepsilon}_{\text{vgd}} \coloneqq \varepsilon_{\text{vgd}} + 12\nu H^2 A \varepsilon_{\text{ex}}$.
- **Local Lipschitz property.** By Lemma 12, the value function is H^2-local Lipschitz w.r.t. the local norm $\pi \mapsto \|\cdot\|_{L^2(\mu^\pi),1}$.
- **Diameter bound.** We have that $\|\pi' - \tilde{\pi}\|_{L^2(\mu^\pi),1} \leq \max_{p,q\in\Delta(\mathcal{A})} \|p - q\|_1 \leq 2$, for all $\pi, \pi', \tilde{\pi} \in \Delta(\mathcal{A})^{\mathcal{S}}$.

The above imply we are in the setting of Theorem 8 with $\beta = 2\sqrt{A}H^3/\sqrt{\varepsilon_{\text{ex}}}$, $M = H^2, D = 2$, and $\epsilon = \tilde{\epsilon} = \varepsilon$. Thus, with step sizes set according to the statement of Theorem 8, we have that after K iterations:

$$\begin{aligned} V(\pi^{K+1}) - V^\star(\Pi) &\lesssim \frac{\nu^2\sqrt{A}H^3}{\sqrt{\varepsilon_{\text{ex}}}K} + \frac{\nu\sqrt{A}H^3}{\sqrt{\varepsilon_{\text{ex}}}}\sqrt{\varepsilon} + \frac{\sqrt{A}H^3}{\sqrt{\varepsilon_{\text{ex}}}}\varepsilon K + \nu A H^2 \varepsilon_{\text{ex}} + \varepsilon_{\text{vgd}} \\ &\leq \frac{\nu^2\sqrt{A}H^3}{\sqrt{\varepsilon_{\text{ex}}}}\left(\frac{1}{K} + \sqrt{\varepsilon} + \varepsilon K\right) + \nu A H^2 \varepsilon_{\text{ex}} + \varepsilon_{\text{vgd}} \end{aligned}$$

where $\lesssim$ hides only universal constant factors. Now, set $\varepsilon_{\text{ex}} = \left(\frac{1}{K} + \sqrt{\varepsilon} + \varepsilon K\right)^{2/3}$, and use the fact that $(a+b)^{2/3} \leq a^{2/3} + b^{2/3}$ to immediately obtain the stated bound. □

D.4 PMD

The PMD method [Tomar et al., 2020, Xiao, 2022, Lan, 2023] make use of an action regularizer, and more specifically the Bregman divergence w.r.t. the chosen regularizer, which we define below.

Definition 9 (Bregman divergence)**.** Given a convex differentiable regularizer $R\colon \mathbb{R}^{\mathcal{A}} \to \mathbb{R}$, the Bregman divergence w.r.t. R is:

$$B_R(u, v) \coloneqq R(u) - R(v) - \langle \nabla R(v), u - v\rangle .$$

We will make use of the following elementary lemma, which follows from standard arguments; for proof see Sherman et al. [2025].

Lemma 14. *Assume $h\colon \mathbb{R}^A \to \mathbb{R}$ is 1-strongly convex and has L-Lipschitz gradient w.r.t. $\|\cdot\|$. Let $\mu \in \Delta(\mathcal{S})$, and define $R_\mu(\pi) \coloneqq \mathbb{E}_{s\sim\mu}[h(\pi_s)]$. Then*

1. *$B_{R_\mu}(\pi, \tilde{\pi}) = \mathbb{E}_{s\sim\mu} B_R(\pi_s, \tilde{\pi}_s)$.*
2. *R_μ is 1-strongly convex and has an L-Lipschitz gradient w.r.t. $\|\cdot\|_{L^2(\mu),\circ}$.*

Below we restate and prove the guarantee for the PMD method detailed in Algorithm 6.

Theorem (Restatement of Theorem 4). Let Π be a convex policy class that satisfies $(\nu, \varepsilon_{\text{vgd}})$-VGD w.r.t. $\mathcal{M}$. Suppose that PMD (see Algorithm 6) is executed with the L^2 action regularizer. Then, with an appropriate tuning of $\eta, \varepsilon_{\text{ex}}$, we have that the output of PMD satisfies:

$$V(\hat{\pi}) - V^\star(\Pi) = O\left(\frac{\nu^2 A^{3/2} H^3}{K^{2/3}} + \left(\nu + H^2 A K^{1/6}\right)\varepsilon^{1/4}\right).$$

Algorithm 7 Action-value estimation

Input: π

Begin rollout at $s_0 \sim \rho_0$

For each timestep $t = 0, \ldots,$ act $a_t \sim \pi_{s_t}$, and $\begin{cases} \text{continue} & \text{w.p. } \gamma \\ \text{accept } s_t & \text{w.p. } 1-\gamma \end{cases}$

After accepting s_t, sample $a_t \sim \mathrm{Unif}(\mathcal{A})$ and continue the rollout, terminating at each step w.p. $1-\gamma$.

Assume the rollout terminated at iteration T. Define $\widehat{Q}^{\pi}_{s_t} \in \mathbb{R}^{\mathcal{A}}$ by

$$\forall a \in \mathcal{A}: \ \widehat{Q}^{\pi}_{s_t,a} = \mathbb{I}\{a = a_t\} A \sum_{t'=t}^{T} r(s_{t'}, a_{t'}).$$

return $s_t, \widehat{Q}^{\pi}_{s_t}$

Proof of Theorem 4. We now verify a number of conditions that place us in the setup of Assumption 1.

- **Local smoothness.** By Lemma 8 and the definition of $\Pi^{\varepsilon_{\mathrm{ex}}}$, the value function is $(2A^{3/2}H^3/\sqrt{\varepsilon_{\mathrm{ex}}})$ locally smooth w.r.t. the local norm $\pi \mapsto \|\cdot\|_{L^2(\mu^\pi),2}$.
- **VGD condition for $\Pi^{\varepsilon_{\mathrm{ex}}}$.** By Lemma 9, we have that $\Pi^{\varepsilon_{\mathrm{ex}}}$ satisfies $(\nu, \tilde{\varepsilon}_{\mathrm{vgd}})$ with $\tilde{\varepsilon}_{\mathrm{vgd}} \coloneqq \varepsilon_{\mathrm{vgd}} + 12\nu H^2 A \varepsilon_{\mathrm{ex}}$.
- **Local Lipschitz property.** By Lemma 12 and the fact that $\left\|Q^{\pi}_s\right\|_2 \le \sqrt{A}H$ for all π, s, the value function is $(\sqrt{A}H^2)$-local Lipschitz w.r.t. the local norm $\pi \mapsto \|\cdot\|_{L^2(\mu^\pi),2}$.
- **Diameter bound.** We have that $\|\pi' - \tilde{\pi}\|_{L^2(\mu^\pi),2} \le \max_{p,q\in\Delta(\mathcal{A})} \|p-q\|_2 \le 2$, for all $\pi, \pi', \tilde{\pi} \in \Delta(\mathcal{A})^{\mathcal{S}}$.
- **Regularizer smoothness.** The euclidean action norm $R(p) = \frac{1}{2}\|p\|_2^2$ is 1-smooth.

The above imply we are in the setting of Theorem 10 with $\beta = 2A^{3/2}H^3/\sqrt{\varepsilon_{\mathrm{ex}}}$, $M = \sqrt{A}H^2, D = 2, L = 1$. Thus, setting $\eta = \sqrt{\varepsilon_{\mathrm{ex}}}/(2H^3A^{3/2})$, we have $c_1 = D + \eta M = O(1)$, and the guarantee that ($\lesssim$ suppresses constant numerical factors):

$$\begin{aligned} V(\pi^{K+1}) - V^\star(\Pi) &\lesssim \frac{\nu^2}{\eta K} + \nu\sqrt{\varepsilon} + \frac{\varepsilon^{1/4}}{\sqrt{\eta}} + \varepsilon_{\mathrm{vgd}} \\ &\lesssim \frac{\nu^2 A^{3/2} H^3}{\sqrt{\varepsilon_{\mathrm{ex}}} K} + \nu\sqrt{\varepsilon} + \frac{\sqrt{H^3 A^{3/2}}}{{\varepsilon_{\mathrm{ex}}}^{1/4}} \varepsilon^{1/4} + \nu A H^2 \varepsilon_{\mathrm{ex}} + \varepsilon_{\mathrm{vgd}} \\ &\lesssim \frac{\nu^2 A^{3/2} H^3}{\sqrt{\varepsilon_{\mathrm{ex}}} K} + \varepsilon^{1/4}\left(\nu + \frac{H^2 A}{{\varepsilon_{\mathrm{ex}}}^{1/4}}\right) + \nu A H^2 \varepsilon_{\mathrm{ex}} + \varepsilon_{\mathrm{vgd}}. \end{aligned}$$

Choosing $\varepsilon_{\mathrm{ex}} = K^{-2/3}$, we immediately obtain the stated bound. □

E Sample complexity upper bounds

In this section, we demonstrate how our iteration complexity upper bounds may be translated to sample complexity upper bounds. The sampling scheme Algorithm 7 we employ to estimate the full gradient step is based on importance sampling and in itself is fairly standard. A similar algorithm can be found in e.g., Agarwal et al. [2021]. Throughout this section we adopt the assumption that $\gamma \le 1/2$ in sake of simplified presentation. In terms of the effective horizon this implies $H \ge 2$, which is the interesting regime. The first lemma given below, provides the connection between optimizing the empirical and population objectives.

Lemma 15. *Let $\widetilde{\Pi}$ be a policy class and suppose $\gamma \le 1/2$. Assume $\mathfrak{D}\colon \mathbb{R}^{\mathcal{A}} \times \mathbb{R}^{\mathcal{A}} \to \mathbb{R}_+$ is a non-negative function that satisfies:*

Algorithm 8 SDPO in the learning letup

Input: $K \geq 1, N \geq 1, \eta > 0, \varepsilon_{\text{ex}} > 0, \varepsilon_{\text{erm}} > 0, \Pi \in \Delta(\mathcal{A})^{\mathcal{S}}$, and action norm $\|\cdot\|_\circ : \mathbb{R}^{\mathcal{A}} \to \mathbb{R}$.
Initialize $\pi^1 \in \Pi^{\varepsilon_{\text{ex}}}$
for $k = 1$ **to** K **do**
  Rollout π^k for N episodes via Algorithm 7, obtain $\mathcal{D}_k = \left\{ s_i^k, \widehat{Q}_{s_i^k}^k \right\}_{i=1}^N$.
  Update $\pi^{k+1} \leftarrow \arg\min^{\varepsilon_{\text{erm}}}_{\pi \in \Pi^{\varepsilon_{\text{ex}}}} \left\{ \widehat{\Phi}_k(\pi) := \frac{1}{N} \sum_{s \in \mathcal{D}_k} \left\langle H\widehat{Q}_s^k, \pi_s \right\rangle + \frac{1}{2\eta} \left\| \pi_s - \pi_s^k \right\|_\circ^2 \right\}$
end for
return $\hat{\pi} := \pi^{K+1}$

- *Boundedness:* $D \geq \mathfrak{D}(p, q)$ *for all* $p, q \in \Delta(\mathcal{A})$.
- *Lipschitz continuity w.r.t. the* 1*-norm:* $|\mathfrak{D}(p, p_0) - \mathfrak{D}(q, p_0)| \leq L \|p - q\|_1$ *for all* $p, q, p_0 \in \Delta(\mathcal{A})$.

Let $\pi^1 \in \widetilde{\Pi}$*, suppose* $\pi^{k+1} \in \widetilde{\Pi}$ *satisfy for all* $k \in [K]$*, for a given learning rate* $0 < \eta \leq 1$*,*

$$\pi^{k+1} \in \arg\min^{\varepsilon_{\text{erm}}}_{\pi \in \widetilde{\Pi}} \left\{ \widehat{\Phi}_k(\pi) := \frac{1}{N} \sum_{s \in \mathcal{D}_k} \left\langle H\widehat{Q}_s^k, \pi_s \right\rangle + \frac{1}{\eta} \mathfrak{D}\left(\pi_s, \pi_s^k\right) \right\}, \tag{29}$$

where $\mathcal{D}_k$ *are a (state, action-value) datasets of size* N *obtained by invoking Algorithm 7. Then, for any* $\delta > 0$*, w.p.* $\geq 1 - \delta$*, it holds that for all* $k \in [K]$*,*

$$\pi^{k+1} \in \arg\min^{\varepsilon}_{\pi \in \widetilde{\Pi}} \left\{ \Phi_k(\pi) := \mathbb{E}_{s \sim \mu^k} \left[\left\langle HQ_s^k, \pi_s \right\rangle + \frac{1}{\eta} \mathfrak{D}\left(\pi_s, \pi_s^k\right) \right] \right\}, \tag{30}$$

where $\varepsilon = \varepsilon_{\text{erm}} + \varepsilon_{\text{gen}}$*,* $\varepsilon_{\text{gen}} := \frac{C_0 A H^2 D}{\eta} \sqrt{\frac{\log \frac{KNN(\varepsilon_{\text{net}}, \widetilde{\Pi}, \|\cdot\|_{\infty,1})}{\delta}}{N}}$*,* $C_0 > 0$ *is an absolute numerical constant, and* $\varepsilon_{\text{net}} \geq \frac{C_0 A H^2 D}{6\sqrt{N}(AH^2 \log(2KN/\delta) + L)}$*. Furthermore, the number of time steps of each episode rolled out by Algorithm 7 is* $\leq 2H \log(2KN/\delta)$*.*

We defer the proof of Lemma 15 to Appendix E.3. We now turn to apply the lemma in conjunction with the iteration complexity guarantee of SDPO (Theorem 1) to obtain a sample complexity upper bound for SDPO in the learning setup (Algorithm 8). Afterwards, we present sample complexity upper bounds for the other algorithms in Appendices E.1 and E.2. We note that there are a number of places where we expect the analysis can be tightened subject to future work; primarily, the greedy exploration required by the current local-smoothness analysis. Hence, the actual rate obtained is not the primary focus of our work. Furthermore, we did not make a notable effort in obtaining optimal dependence on all problem parameters, and expect these can be easily tightened by a more careful choice analysis and choice of algorithm input-parameters.

Theorem 11. *Let* Π *be a convex policy class that satisfies* $(\nu, \varepsilon_{\text{vgd}})$*-VGD w.r.t.* $\mathcal{M}$*, and assume* $\gamma \leq 1/2$*. Then for any* $n \geq 1$*, there exists a choice of parameters* $K, N, \eta, \varepsilon_{\text{ex}}$ *such that Algorithm 8 executed with the* L^1 *action norm guarantees for any* $\delta > 0$*, that w.p.* $\geq 1 - \delta$ *the total number of environment time steps* $\leq n$*, and the output policy satisfies*

$$V(\hat{\pi}) - V^\star(\Pi) = O\left(\frac{\nu^2 A^2 H^7 \sqrt{\log \frac{nC(\Pi)}{\delta}}}{n^{2/15}} + \nu H^3 \sqrt{A} n^{1/30} \sqrt{\varepsilon_{\text{erm}}} + \varepsilon_{\text{vgd}} \right),$$

where $C(\Pi) := \mathcal{N}(\varepsilon_{\text{net}}, \Pi, \|\cdot\|_{\infty,1})$ *is the* ε_{net}*-covering number of* Π *and* $\varepsilon_{\text{net}} = \Omega\left(\frac{1}{\log(n/\delta)n}\right)$.

Proof. Note that for any $p, q \in \Delta(\mathcal{A})$, $\frac{1}{2}\|p - q\|_1^2 \leq 2$, and

$$\begin{aligned}
\frac{1}{2}\|p - p_0\|_1^2 - \frac{1}{2}\|q - p_0\|_1^2 &= \frac{1}{2}\left(\|p - p_0\|_1 + \|q - p_0\|_1\right)\left(\|p - p_0\|_1 - \|q - p_0\|_1\right) \\
&\leq 2\left(\|p - p_0\|_1 - \|q - p_0\|_1\right) \\
&\leq 2\|p - q\|_1 .
\end{aligned}$$

Thus, when executing Algorithm 8 over K iterations, we are in the setting of Lemma 15 with $D = 2, L = 2$. Suppose we run the algorithm for K iterations with $\eta = \sqrt{\varepsilon_{\text{ex}}}/(2H^3\sqrt{A})$, $\varepsilon_{\text{ex}} = H^2/K^{2/3}$, and N (and K) to be chosen later on. For any $\delta > 0$, we have by Lemma 15 that w.p. $\geq 1 - \delta$, for all $k \in [K]$ it holds that:

$$\pi^{k+1} \in \arg\min^{\varepsilon}_{\pi \in \Pi^{\varepsilon_{\text{ex}}}} \left\{ \Phi_k(\pi) \coloneqq \mathbb{E}_{s\sim\mu^k} \left[\left\langle HQ^k_s, \pi_s \right\rangle + \frac{1}{2\eta} \left\| \pi_s - \pi^k_s \right\|_1^2 \right] \right\}, \tag{31}$$

with $\varepsilon = \varepsilon_{\text{erm}} + \varepsilon_{\text{gen}}$, where

$$\varepsilon_{\text{gen}} = O\left(\frac{AH^2}{\eta} \sqrt{\frac{\log \frac{KNC(\Pi)}{\delta}}{N}} \right),$$

$C(\Pi) \coloneqq \mathcal{N}(\varepsilon_{\text{net}}, \Pi, \|\cdot\|_{\infty,1})$, and $\varepsilon_{\text{net}} = \Omega\left(\frac{1}{\log(KN/\delta)\sqrt{N}}\right)$. Now, Eq. (31) implies Algorithm 8 is an instance of the idealized SDPO Algorithm 1 with the error ε defined above. Hence, by Theorem 1, we have that

$$\begin{aligned} V(\hat{\pi}) - V^\star(\Pi) &= O\left(\frac{\nu^2 AH^4}{K^{2/3}} + \nu H^3 \sqrt{A} K^{1/6} \sqrt{\varepsilon} + \varepsilon_{\text{vgd}} \right) \\ &= O\left(\frac{\nu^2 AH^4}{K^{2/3}} + \nu H^3 \sqrt{A} K^{1/6} \sqrt{\varepsilon_{\text{gen}}} + \nu H^3 \sqrt{A} K^{1/6} \sqrt{\varepsilon_{\text{erm}}} + \varepsilon_{\text{vgd}} \right). \end{aligned} \tag{32}$$

We focus on the first two terms to choose K, N as a function of n. Observe:

$$\begin{aligned} \frac{\nu^2 AH^4}{K^{2/3}} + \nu H^3 \sqrt{A} K^{1/6} \sqrt{\varepsilon_{\text{gen}}} &\approx \frac{\nu^2 AH^4}{K^{2/3}} + \nu H^3 \sqrt{A} K^{1/6} \frac{\sqrt{A}H}{\sqrt{\eta}} \frac{\log^{1/4} \frac{KNC(\Pi)}{\delta}}{N^{1/4}} \\ &= \frac{\nu^2 AH^4}{K^{2/3}} + \frac{\nu AH^4 K^{1/6} \iota}{\sqrt{\eta} N^{1/4}} \end{aligned}$$

with $\iota \coloneqq \log^{1/4} \frac{KNC(\Pi)}{\delta}$. Further, by our choice of $\eta, \varepsilon_{\text{ex}}$, $\eta = H/(2H^3\sqrt{A}K^{1/3})$, hence

$$\begin{aligned} \frac{\nu^2 AH^4}{K^{2/3}} + \frac{\nu AH^4 K^{1/6} \iota}{\sqrt{\eta} N^{1/4}} &\approx \frac{\nu^2 AH^4}{K^{2/3}} + \frac{\nu AH^4 K^{1/6} \iota \sqrt{H^3 \sqrt{A} K^{1/3}}}{\sqrt{H} N^{1/4}} \\ &\leq \frac{\nu^2 AH^6}{K^{2/3}} + \frac{\nu A^2 H^5 \iota}{N^{1/4}} K^{1/3}. \end{aligned}$$

Choosing $N = K^4$ gives

$$\frac{\nu^2 AH^6}{K^{2/3}} + \frac{\nu A^2 H^5 \iota}{N^{1/4}} K^{1/3} \lesssim \frac{\nu^2 A^2 H^6 \iota}{K^{2/3}},$$

with $n \leq KN\widetilde{H} = K^5 \widetilde{H}$ with $\widetilde{H} = H \log(2KN/\delta)$ by Lemma 15. Hence $K \geq n^{1/5}/\widetilde{H}^{1/5}$, and

$$\frac{\nu^2 A^2 H^6 \iota}{K^{2/3}} \leq \frac{\nu^2 A^2 H^6 \widetilde{H}^{2/15} \iota}{n^{2/15}}.$$

Now substitute $\widetilde{H}^{2/15} \lesssim H^{1/2} \log^{1/4}(n/\delta)$, and $\iota = \log^{1/4} \frac{KNC(\Pi)}{\delta} \lesssim \log^{1/4} \frac{nC(\Pi)}{\delta}$ to obtain

$$\frac{\nu^2 A^2 H^6 \widetilde{H}^{2/15} \iota}{n^{2/15}} \lesssim \frac{\nu^2 A^2 H^7 \sqrt{\log \frac{nC(\Pi)}{\delta}}}{n^{2/15}}.$$

Finally, using that and plugging our upper bound back into Eq. (32) and using that $K \leq n^{1/5}$:

$$\begin{aligned} V(\hat{\pi}) - V^\star(\Pi) &= O\left(\frac{\nu^2 A^2 H^7 \sqrt{\log \frac{nC(\Pi)}{\delta}}}{n^{2/15}} + \nu H^3 \sqrt{A} K^{1/6} \sqrt{\varepsilon_{\text{erm}}} + \varepsilon_{\text{vgd}} \right) \\ &= O\left(\frac{\nu^2 A^2 H^7 \sqrt{\log \frac{nC(\Pi)}{\delta}}}{n^{2/15}} + \nu H^3 \sqrt{A} n^{1/30} \sqrt{\varepsilon_{\text{erm}}} + \varepsilon_{\text{vgd}} \right), \end{aligned}$$

with $\varepsilon_{\text{net}} = \Omega\left(\frac{1}{\log(KN/\delta)\sqrt{N}}\right) = \Omega\left(\frac{1}{\log(n/\delta)\sqrt{N}}\right) = \Omega\left(\frac{1}{\log(n/\delta)n^{2/5}}\right)$. □

E.1 CPI and DA-CPI

In this section we present the learning versions of CPI (Algorithm 9) and DA-CPI (Algorithm 10) and their sample complexity guarantees. We first state two lemmas which play the same role of Lemma 15. The proofs follow from identical arguments as those of Lemma 15, and are thus omitted.

Algorithm 9 CPI in the learning setup

input: Initial policy $\pi^1 \in \Pi$, $\{\eta_k\}_{k=1}^K$, $N \geq 1$, $\varepsilon_{\text{erm}} > 0$
for $k = 1, 2, \ldots, K$ **do**
  Rollout π^k for N episodes via Algorithm 7, obtain $\mathcal{D}_k = \left\{ s_i^k, \widehat{Q}_{s_i^k}^k \right\}_{i=1}^N$.
  Update $\tilde{\pi}^{k+1} \leftarrow \arg\min_{\pi \in \Pi}^{\varepsilon_{\text{erm}}} \left\{ \widehat{\Phi}_k(\pi) := \frac{1}{N} \sum_{s \in \mathcal{D}_k} \left\langle H\widehat{Q}_s^k, \pi_s - \pi_s^k \right\rangle \right\}$
  Set $\pi^{k+1} = (1 - \eta_k)\pi^k + \eta_k \tilde{\pi}^{k+1}$
end for
return $\hat{\pi} = \pi^{K+1}$

Algorithm 10 DA-CPI in the learning setup

input: $\eta_1, \ldots, \eta_K > 0$; $N \geq 1$, $\varepsilon_{\text{ex}} > 0$, $\varepsilon_{\text{erm}} > 0$, action norm $\|\cdot\|_\circ$.
for $k = 1, \ldots, K$ **do**
  Rollout π^k for N episodes via Algorithm 7, obtain $\mathcal{D}_k = \left\{ s_i^k, \widehat{Q}_{s_i^k}^k \right\}_{i=1}^N$.
  Update $\tilde{\pi}^{k+1} \leftarrow \arg\min_{\pi \in \Pi^{\varepsilon_{\text{ex}}}}^{\varepsilon_{\text{erm}}} \left\{ \widehat{\Phi}_k(\pi) := \frac{1}{N} \sum_{s \in \mathcal{D}_k} \left\langle \widehat{Q}_s^k, \pi_s \right\rangle \right\}$
  Rollout π^k for another N episodes via Algorithm 7, obtain $\widetilde{\mathcal{D}}_k$.
  Update $\pi^{k+1} \leftarrow \arg\min_{\pi \in \Pi^{\varepsilon_{\text{ex}}}}^{\varepsilon_{\text{erm}}} \left\{ \widehat{\psi}_k(\pi) := \frac{1}{N} \sum_{s \in \widetilde{\mathcal{D}}_k} \left\| \pi_s - ((1 - \eta_k)\pi_s^k + \eta_k \tilde{\pi}_s^{k+1}) \right\|_\circ^2 \right\}$
end for
return $\hat{\pi} = \pi^{K+1}$

Lemma 16. *For $\gamma \leq 1/2$, upon execution of Algorithm 9, for any $\delta > 0$, w.p. $\geq 1 - \delta$, it holds that for all $k \in [K]$,*

$$\tilde{\pi}^{k+1} \in \arg\min_{\pi \in \Pi}^{\varepsilon} \mathbb{E}_{s \sim \mu^k} \left[\left\langle HQ_s^k, \pi_s - \pi_s^k \right\rangle \right],$$

where $\varepsilon = \varepsilon_{\text{erm}} + \varepsilon_{\text{gen}}$, $\varepsilon_{\text{gen}} := C_0 A H^2 D \sqrt{\frac{\log \frac{KNN(\varepsilon_{\text{net}}, \widetilde{\Pi}, \|\cdot\|_{\infty,1})}{\delta}}{N}}$, $C_0 > 0$ is an absolute numerical constant, and $\varepsilon_{\text{net}} \geq \frac{C_0}{6\sqrt{N}(\log(2KN/\delta)}$. Furthermore, the number of time steps of each episode rolled out by Algorithm 7 is $\leq 2H \log(2KN/\delta)$.

Lemma 17. *For $\gamma \leq 1/2$, upon execution of Algorithm 10, for any $\delta > 0$, w.p. $\geq 1 - \delta$, it holds that for all $k \in [K]$,*

$$\tilde{\pi}^{k+1} \in \arg\min_{\pi \in \Pi^{\varepsilon_{\text{ex}}}}^{\varepsilon} \mathbb{E}_{s \sim \mu^k} \left[\left\langle HQ_s^k, \pi_s - \pi_s^k \right\rangle \right]$$
$$\pi^{k+1} \in \arg\min_{\pi \in \Pi^{\varepsilon_{\text{ex}}}}^{\varepsilon} \mathbb{E}_{s \sim \mu^k} \left\| \pi_s - ((1 - \eta_k)\pi_s^k + \eta_k \tilde{\pi}_s^{k+1}) \right\|_1^2,$$

where $\varepsilon = \varepsilon_{\text{erm}} + \varepsilon_{\text{gen}}$, $\varepsilon_{\text{gen}} := C_0 A H^2 D \sqrt{\frac{\log \frac{KNN(\varepsilon_{\text{net}}, \widetilde{\Pi}, \|\cdot\|_{\infty,1})}{\delta}}{N}}$, $C_0 > 0$ is an absolute numerical constant, and $\varepsilon_{\text{net}} \geq \frac{C_0}{6\sqrt{N}(\log(2KN/\delta)}$. Furthermore, the number of time steps of each episode rolled out by Algorithm 7 is $\leq 2H \log(2KN/\delta)$.

We now give the sample complexity guarantees of CPI and DA-CPI; for simplicity, we present the bounds assuming $\varepsilon_{\text{erm}} = 0$.

Theorem 12. *Let Π be a policy class that satisfies $(\nu, \varepsilon_{\text{vgd}})$-VGD w.r.t. $\mathcal{M}$, and assume $\gamma \leq 1/2$. Then for any $n \geq 1$, there exists a choice of parameters $K, N, \{\eta_k\}$ such that Algorithm 9 executed*

with $\varepsilon_{\text{erm}} = 0$ and the L^1 action norm, guarantees for any $\delta > 0$, that w.p. $\geq 1 - \delta$ the total number of environment time steps $\leq n$, and the output policy satisfies

$$V(\hat{\pi}) - V^\star(\Pi) = O\left(\frac{\nu^2 A H^4 \log(nC(\Pi)/\delta)}{n^{1/3}} + \varepsilon_{\text{vgd}}\right),$$

where $C(\Pi) \coloneqq \mathcal{N}(\varepsilon_{\text{net}}, \Pi, \|\cdot\|_{\infty,1})$ is the ε_{net}-covering number of Π and $\varepsilon_{\text{net}} = \Omega\left(\frac{1}{\log(n/\delta)n}\right)$.

Proof. By Lemma 16, we have that w.p. $\geq 1 - \delta$, for all $k \in [K]$ it holds that:

$$\pi^{k+1} \in \arg\min^{\varepsilon}_{\pi \in \Pi} \left\{\Phi_k(\pi) \coloneqq \mathbb{E}_{s\sim\mu^k}\left[\left\langle HQ^k_s, \pi_s - \pi^k_s\right\rangle\right]\right\},$$

with $\varepsilon = \varepsilon_{\text{erm}} + \varepsilon_{\text{gen}}$, where

$$\varepsilon_{\text{gen}} = O\left(AH^2\sqrt{\frac{\log\frac{KNC(\Pi)}{\delta}}{N}}\right),$$

$C(\Pi) \coloneqq \mathcal{N}(\varepsilon_{\text{net}}, \Pi, \|\cdot\|_{\infty,1})$, and $\varepsilon_{\text{net}} = \Omega(\frac{1}{\log(KN/\delta)\sqrt{N}})$. Hence, Algorithm 9 is an instance of the idealized CPI Algorithm 2 with the error ε defined above. Now, by Theorem 2, we have that

$$\begin{aligned} V(\hat{\pi}) - V^\star(\Pi) &= O\left(\frac{\nu^2 H^3}{K} + \nu\varepsilon + \varepsilon_{\text{vgd}}\right) \\ &= O\left(\frac{\nu^2 H^3}{K} + \nu A H^2\sqrt{\frac{\log\frac{KNC(\Pi)}{\delta}}{N}} + \varepsilon_{\text{vgd}}\right). \end{aligned}$$

Choosing $N = K^2$, and noting that $n \lesssim N^{3/2} H \log(n/\delta)$, we obtain

$$V(\hat{\pi}) - V^\star(\Pi) = O\left(\frac{\nu^2 A H^4 \log(nC(\Pi)/\delta)}{n^{1/3}} + \varepsilon_{\text{vgd}}\right),$$

which completes the proof. □

Theorem 13. *Let Π be a policy class that satisfies $(\nu, \varepsilon_{\text{vgd}})$-VGD w.r.t. $\mathcal{M}$, and assume $\gamma \leq 1/2$. Then for any $n \geq 1$, there exists a choice of parameters $K, N, \varepsilon_{\text{ex}}, \{\eta_k\}$ such that Algorithm 10 executed with with $\varepsilon_{\text{erm}} = 0$ and the L^1 action norm, guarantees for any $\delta > 0$, that w.p. $\geq 1 - \delta$ the total number of environment time steps $\leq n$, and the output policy satisfies*

$$V(\hat{\pi}) - V^\star(\Pi) = O\left(\frac{\nu^2 A^2 H^5 \sqrt{\log(nC(\Pi)/\delta)}}{n^{2/15}} + \varepsilon_{\text{vgd}}\right),$$

where $C(\Pi) \coloneqq \mathcal{N}(\varepsilon_{\text{net}}, \Pi, \|\cdot\|_{\infty,1})$ is the ε_{net}-covering number of Π and $\varepsilon_{\text{net}} = \Omega\left(\frac{1}{\log(n/\delta)n}\right)$.

Proof. By Lemma 17, we have that w.p. $\geq 1 - \delta$, sub-optimality ε holds for all $k \in [K]$ and for both update steps with $\varepsilon = \varepsilon_{\text{erm}} + \varepsilon_{\text{gen}}$, where

$$\varepsilon_{\text{gen}} = O\left(AH^2\sqrt{\frac{\log\frac{KNC(\Pi)}{\delta}}{N}}\right),$$

$C(\Pi) \coloneqq \mathcal{N}(\varepsilon_{\text{net}}, \Pi, \|\cdot\|_{\infty,1})$, and $\varepsilon_{\text{net}} = \Omega(\frac{1}{\log(KN/\delta)\sqrt{N}})$. Hence, Algorithm 10 is an instance of the idealized DA-CPI Algorithm 2 with the error ε defined above. Now, by Theorem 2, we have that

$$V(\hat{\pi}) - V^\star(\Pi) = O\left(\nu^2 A H^3\left(\frac{1}{K^{2/3}} + \varepsilon^{1/3} + \varepsilon^{2/3}K^{2/3}\right) + \varepsilon_{\text{vgd}}\right).$$

We focus on the first two terms; when balancing them, the third will be of the same order. Let $\iota \coloneqq \sqrt{\log(KNC(\Pi)/\delta)}$, and we have:

$$\begin{aligned}
\frac{1}{K^{2/3}} + \varepsilon^{1/3} &\lesssim \frac{1}{K^{2/3}} + \frac{(AH^2\iota)^{1/3}}{N^{1/6}} \\
&\lesssim \frac{(AH^2\iota)^{1/3}}{K^{2/3}} && (N = K^4) \\
&\lesssim \frac{AH^2\iota}{n^{2/15}}. && (n \lesssim K^5 H \log(KN/\delta))
\end{aligned}$$

This implies

$$V(\hat{\pi}) - V^\star(\Pi) = O\left(\frac{\nu^2 A^2 H^5 \iota}{n^{2/15}} + \varepsilon_{\mathrm{vgd}}\right),$$

which completes the proof. □

E.2 PMD

PMD in the learning setup is given in Algorithm 11. Since L^2-PMD and L^2-SDPO coincide (they are the exact same algorithm), the sample complexity of PMD with the Euclidean action regularizer follows form arguments identical to those given for SDPO in the proof of Theorem 11, but with the L^2-action norm instead of the L^1-norm. This leads to slightly worse dependence on the action set cardinality A, but otherwise to the same guarantee.

Theorem 14. *Let Π be a convex policy class that satisfies $(\nu, \varepsilon_{\mathrm{vgd}})$-VGD w.r.t. $\mathcal{M}$, and assume $\gamma \leq 1/2$. Then for any $n \geq 1$, there exists a choice of parameters $K, N, \eta, \varepsilon_{\mathrm{ex}}$ such that Algorithm 11 executed with $\varepsilon_{\mathrm{erm}} = 0$ and the L^2 action regularizer guarantees for any $\delta > 0$, that w.p. $\geq 1 - \delta$ the total number of environment time steps $\leq n$, and the output policy satisfies*

$$V(\hat{\pi}) - V^\star(\Pi) = O\left(\frac{\nu^2 A^3 H^7 \sqrt{\log \frac{nC(\Pi)}{\delta}}}{n^{2/15}} + \varepsilon_{\mathrm{vgd}}\right),$$

where $C(\Pi) \coloneqq \mathcal{N}(\varepsilon_{\mathrm{net}}, \Pi, \|\cdot\|_{\infty,1})$ is the $\varepsilon_{\mathrm{net}}$-covering number of Π and $\varepsilon_{\mathrm{net}} = \Omega\left(\frac{1}{\log(n/\delta)n}\right)$.

Notably, the analysis of L^2-PMD through the SDPO perspective gives better sample complexity than we would have obtained through the Bregman-proximal method analysis Theorem 10; roughly speaking this is the case because analysis based on the Bregman divergence hinges on approximate optimality conditions rather than sub-optimality in function values. This leads to dependence of ${\varepsilon_{\mathrm{gen}}}^{1/4}$ rather than $\sqrt{\varepsilon_{\mathrm{gen}}}$, which further leads to inferior sample complexity. For action regularizers other than L^2, sample complexity of PMD may be derived again using similar arguments to those of Theorem 11 but now combined with Theorem 10. A little more care is needed in the choice of parameters, as smoothness of the action regularizer (needed both in Theorem 10 and in Lemma 15) is commonly inversely related to $\varepsilon_{\mathrm{ex}}$. As a result, with the currently known techniques for the iteration complexity upper bound, the sample complexity upper bounds for essentially any action regularizer other than L^2, will be worse than those of the L^2 case w.r.t. dependence on both n and A.

E.3 Proof of Lemma 15

Lemma 18. *Algorithm 7 returns $s_t, \widehat{Q}^\pi_{s_t}$ that satisfy (i) $s_t \sim \mu^\pi$; (ii) $\mathbb{E}\left[\widehat{Q}^\pi_{s_t} \mid s_t\right] = Q^\pi_{s_t}$. Furthermore, for any $\delta > 0$, w.p. $\geq 1 - \delta$, the algorithm terminates after no more than $\frac{2}{1-\gamma} \log \frac{2}{\delta}$ time steps and it holds that $\left|\widehat{Q}^\pi_{s_t,a_t}\right| \leq HA \log(2/\delta)$.*

Proof of Lemma 18. First, note that (i) follows directly from the definition of the discounted occupancy measure Eq. (12). Indeed, by definition of Algorithm 7, for any $s \in \mathcal{S}$:

$$\Pr(s \text{ is accepted by Algorithm 7 on step } t) = \gamma^t (1-\gamma) \Pr(s_t = s \mid \pi, s_0 \sim \rho_0)$$

$$\implies \Pr(s \text{ is accepted by Algorithm 7}) = (1-\gamma) \sum_{t=0}^{\infty} \gamma^t \Pr(s_t = s \mid \pi, s_0 \sim \rho_0) = \mu^\pi(s).$$

Algorithm 11 PMD in the learning setup

Input: $K \geq 1, N \geq 1, \eta > 0, \varepsilon_{\text{ex}} > 0, \varepsilon_{\text{erm}} > 0, \Pi \in \Delta(\mathcal{A})^{\mathcal{S}}$, and action regularizer $R\colon \mathbb{R}^{\mathcal{A}} \to \mathbb{R}$.

Initialize $\pi^1 \in \Pi^{\varepsilon_{\text{ex}}}$
for $k = 1$ **to** K **do**
  Rollout π^k for N episodes via Algorithm 7, obtain $\mathcal{D}_k = \left\{ s_i^k, \widehat{Q}^k_{s_i^k} \right\}_{i=1}^N$.
  Update $\pi^{k+1} \leftarrow \arg\min^{\varepsilon_{\text{erm}}}_{\pi \in \Pi^{\varepsilon_{\text{ex}}}} \left\{ \widehat{\Phi}_k(\pi) := \frac{1}{N} \sum_{s \in \mathcal{D}_k} \left\langle H\widehat{Q}^k_s, \pi_s \right\rangle + \frac{1}{\eta} B_R \pi_s, \pi^k_s \right\}$
end for
return $\hat{\pi} := \pi^{K+1}$

Further, let **alg** denote Algorithm 7, and then

$$\begin{aligned}
\mathbb{E}_{\mathbf{alg}}\left[\widehat{Q}^{\pi}_{s_t,a_t} \mid s_t, a_t\right] &= A\mathbb{E}_{\mathbf{alg}}\left[\sum_{t'=t}^{T} r(s_{t'}, a_{t'}) \mid s_t, a_t\right] \\
&= A\sum_{t'=t}^{\infty} \gamma^{t-t'} \mathbb{E}_{\pi}\left[r(s_{t'}, a_{t'}) \mid s_t, a_t\right] \\
&= A\sum_{t'=t}^{\infty} \gamma^{t-t'} \mathbb{E}_{\pi}\left[r(x_h, u_h) \mid x_0 = s_t, u_0 = a_t\right],
\end{aligned}$$

where in the last expression, expectation is w.r.t. $x_{h+1} \sim \mathbb{P}(\cdot|x_h, u_h)$ for $h \geq 1$, and $u_h \sim \pi(\cdot|x_h)$ for $h \geq 2$. Now,

$$\begin{aligned}
\sum_{t'=t}^{\infty} \gamma^{t-t'} \mathbb{E}_{\pi}\left[r(x_h, u_h) \mid x_0 = s_t, u_0 = a_t\right] &= \sum_{h=0}^{\infty} \gamma^h \mathbb{E}_{\pi}\left[r(x_h, u_h) \mid x_0 = s_t, u_0 = a_t\right] \\
&= \mathbb{E}_{\pi}\left[\sum_{h=0}^{\infty} \gamma^h r(x_h, u_h) \mid x_0 = s_t, u_0 = a_t\right] \\
&= Q^{\pi}_{s_t,a_t},
\end{aligned}$$

hence for all $a \in \mathcal{A}$,

$$\mathbb{E}_{\mathbf{alg}}\left[\widehat{Q}^{\pi}_{s_t,a} \mid s_t\right] = \Pr(a_t = a)\mathbb{E}_{\mathbf{alg}}\left[\widehat{Q}^{\pi}_{s_t,a_t} \mid s_t, a_t = a\right] = \Pr(a_t = a)AQ^{\pi}_{s_t,a} = Q^{\pi}_{s_t,a},$$

which proves (ii).

For the second part, observe that the acceptance event occurs w.p.$1 - \gamma$ at each time step, therefore the probability for acceptance to not occur in the first t times steps is γ^t, and hence probability of acceptance by time t is $1 - \gamma^t$. We have $\gamma^t \leq e^{-(1-\gamma)t} \leq \delta$ for $t \geq \frac{1}{1-\gamma} \log \frac{1}{\delta}$, hence, w.p. $\geq 1 - \delta/2$ acceptance occurs before time step $t_\delta := \frac{1}{1-\gamma} \log \frac{2}{\delta}$. Same goes for the termination event, hence, the episode terminates after $2t_\delta$ time steps w.p. $\geq 1 - \delta$.

Finally, the bound on $\left|\widehat{Q}^{\pi}_{s_t,a_t}\right|$ follows from the termination event and definition of $\widehat{Q}^{\pi}_{s_t,a_t}$ in the algorithm. □

Our proof makes use of the notion of sub-exponential norm [Vershynin, 2018] of a random variable X:

$$\|X\|_{\psi_1} := \inf\left\{\alpha > 0 : \mathbb{E}e^{|X|/\alpha} \leq 2\right\}. \tag{33}$$

Lemma 19. *Assume T is a geometric random variable, $T \sim \text{Geom(p)}$, i.e., $\Pr(T = t) = (1-p)^{t-1}p$ for $1 \leq t \in \mathbb{N}$. Then, for $q := \max\{p, 1-p\}$:*

$$\|T\|_{\psi_1} \leq 1/\ln\left(1 + \frac{1-q}{4q}\right).$$

Proof. We have:

$$\begin{aligned}\mathbb{E}e^{T/\alpha} &= \sum_{t=1}^{\infty}(1-p)^{t-1}pe^{t/\alpha} \\ &= pe^{1/\alpha}\sum_{t=1}^{\infty}(1-p)^{t-1}e^{(t-1)/\alpha} \\ &= pe^{1/\alpha}\sum_{t=0}^{\infty}\left((1-p)e^{1/\alpha}\right)^{t}\end{aligned}$$

Let $q \coloneqq \max\{p, 1-p\}$, and set $\alpha = 1/\ln\left(1+\frac{1-q}{4q}\right)$, then,

$$e^{1/\alpha} = 1 + \frac{1-q}{4q}.$$

Now,

$$(1-p)e^{1/\alpha} = 1 - p + (1-p)\frac{1-q}{4q} \leq 1 - p + \frac{1-q}{4} \leq 1 - p + \frac{p}{4},$$

hence

$$\sum_{t=0}^{\infty}\left((1-p)e^{1/\alpha}\right)^{t} \leq \frac{1}{1-\left(1-p+\frac{p}{4}\right)} = \frac{4}{3p}.$$

Combining with our previous derivation, we obtain

$$\mathbb{E}e^{T/\alpha} \leq pe^{1/\alpha}\frac{4}{3p} = \frac{4}{3}\left(1+\frac{1-q}{4q}\right) \leq \frac{4}{3}\left(1+\frac{1}{4}\right) = \frac{5}{3} \leq 2.$$

□

Lemma 20. *Assume X is a random variable that satisfies $\|X\|_{\psi_1} \leq R$. Then for N independent samples of $X_1, \ldots, X_N$, we have for any $\epsilon \leq R$:*

$$\Pr\left(\left|\frac{1}{N}\sum_{i=1}^{N}T_i - \mathbb{E}T\right| \geq \epsilon\right) \leq 2e^{-c\frac{N\epsilon^2}{R^2}},$$

where $c > 0$ is an absolute numerical constant.

Centering only costs an absolute constant factor [Vershynin, 2018], thus $\|X - \mathbb{E}x\|_{\psi_1} \leq \tilde{R}$ for $\tilde{R} = c_0 R$. Now, the Bernstein inequality for sums of independent sub-exponential random variables (Theorem 2.8.1 of Vershynin, 2018) yields:

$$\begin{aligned}\Pr\left(\left|\frac{1}{N}\sum_{i=1}^{N}X_i - \mathbb{E}X\right| \geq \epsilon\right) &\leq 2\exp\left[-c_1\min\left(\frac{N^2\epsilon^2}{N\tilde{R}^2}, \frac{N\epsilon}{\tilde{R}}\right)\right] \\ &= 2\exp\left[-c_1 N\min\left(\frac{\epsilon^2}{\tilde{R}^2}, \frac{\epsilon}{\tilde{R}}\right)\right] \\ &= 2\exp\left[-(c_1/c_0)\frac{\epsilon^2 N}{R^2}\right]\end{aligned}$$

where we use the assumption that $\epsilon \leq R \leq \tilde{R}$.

Lemma 21 (Empirical objective concentration)**.** *For a given fixed k and a given fixed policy π, we have that for any $\delta' > 0$, w.p. $\geq 1 - \delta'$ it holds that:*

$$\left|\Phi_k(\pi) - \widehat{\Phi}_k(\pi)\right| \leq \frac{CAH^2D}{\eta}\sqrt{\frac{\log\frac{2}{\delta'}}{N}},$$

where $C > 0$ is a universal numerical constant and $\Phi_k, \widehat{\Phi}_k$ are defined in Eqs. (29) *and* (30) *of Lemma 18.*

Proof. Denote:

$$\hat{\ell}^k(\pi; s) \coloneqq H\left\langle \widehat{Q}_s^k, \pi_s \right\rangle + \frac{1}{\eta}\mathfrak{D}\left(\pi_s, \pi_s^k\right),$$

and note that by Lemma 18, we have

$$\Phi_k(\pi) = \mathbb{E}_{(s,\widehat{Q}_s^k)\sim \text{sampler}(\pi^k)}\left[\hat{\ell}^k(\pi; s)\right],$$

where "sampler" denotes Algorithm 7. Now, for $(s, \widehat{Q}_s^k) \sim \text{sampler}(\pi^k)$, we consider the RV

$$X = \left\langle \widehat{Q}_s^k, \pi_s \right\rangle + \frac{1}{\eta}\mathfrak{D}(\pi_s, \pi_s^k).$$

The divergence term is bounded by D/η, hence $\left\|\frac{1}{\eta}\mathfrak{D}(\pi_s, \pi_s^k)\right\|_{\psi_1} \leq D/\eta$ follows immediately. Further, the RV $\frac{1}{A}\widehat{Q}_{s,a}^k$ for the action a accepted in Algorithm 7 is dominated by a geometric RV $T \sim \text{Geom}(\text{p} = 1-\gamma)$, therefore $\left\|H\widehat{Q}_{s,a}^k\right\|_{\psi_1} \leq HA\left\|T\right\|_{\psi_1}$. By Lemma 19 and our assumption that $\gamma \leq 1/2$,

$$\left\|T\right\|_{\psi_1} \leq \frac{1}{\ln\left(1 + \frac{1-\gamma}{4\gamma}\right)} \leq \frac{1}{\frac{1-\gamma}{8\gamma}} \leq \frac{8}{1-\gamma} = 8H.$$

Now, again using that $\left\|\cdot\right\|_{\psi_1}$ is a norm, we obtain:

$$\left\|H\left\langle \widehat{Q}_s^k, \pi_s \right\rangle + \frac{1}{\eta}\mathfrak{D}(\pi_s, \pi_s^k)\right\|_{\psi_1} \leq D/\eta + 8H^2A.$$

Now, by Lemma 20, we have

$$\Pr\left(\left|\widehat{\Phi}_k(\pi) - \Phi_k(\pi)\right| \geq \epsilon\right) \leq 2e^{-c'\frac{N\epsilon^2}{A^2H^4 + D^2/\eta^2}} \leq 2e^{-c'\frac{N\epsilon^2\eta^2}{A^2H^4D^2}}.$$

for an appropriate universal constant c'. Letting $\delta = 2e^{-c'\frac{N\epsilon^2\eta^2}{A^2H^4D^2}}$, the result follows. □

We are now ready for the proof of Lemma 15.

Proof of Lemma 15. Consider the "good event" described next. Let $\delta > 0$ and denote

$$\widetilde{H} \coloneqq H\log(2KN/\delta),$$
$$\varepsilon_{\text{net}} \coloneqq \frac{CAH^2D}{\sqrt{N}(A\widetilde{H}H + L/\eta)},$$

where C is specified by Lemma 21. Further, let $\text{Net}(\varepsilon_{\text{net}}, \widetilde{\Pi}) \coloneqq \text{Net}(\varepsilon_{\text{net}}, \widetilde{\Pi}, \|\cdot\|_{\infty,1})$ be an ε_{net}-cover of $\widetilde{\Pi}$ w.r.t. $\|\cdot\|_{\infty,1}$ of size $\mathcal{N}(\varepsilon_{\text{net}}, \widetilde{\Pi}) \coloneqq \mathcal{N}(\varepsilon_{\text{net}}, \widetilde{\Pi}, \|\cdot\|_{\infty,1})$. Consider the following events:

1. For all $k \in [K], i \in [N] : \widehat{Q}_{s_i^k, a_i^k}^k \leq \widetilde{H}A$, and the corresponding episode length $\leq \widetilde{H}$
2. For all $k \in [K]$:

$$\forall \pi \in \text{Net}(\varepsilon_{\text{net}}, \widetilde{\Pi}): \quad |\widehat{\Phi}_k(\pi) - \Phi_k(\pi)| \leq \frac{CAH^2D}{\eta}\sqrt{\frac{\log\frac{2KN\mathcal{N}(\varepsilon_{\text{net}}, \widetilde{\Pi})}{\delta}}{N}}$$

By Lemma 18 and the union bound, event (1) holds w.p. $\geq 1 - \delta/2$, and by Lemma 21 and the union bound, event (2) holds w.p. $\geq 1 - \delta/2$. Hence, the good event holds w.p. $\geq 1 - \delta$. Proceeding, we assume the good event holds. We have that for any π, π',

$$\begin{aligned}
|\Phi_k(\pi) - \Phi_k(\pi')| &= \left|\mathbb{E}_{s\sim\mu^{\pi^k}}\left[H\left\langle Q_s^k, \pi_s - \pi_s' \right\rangle + \frac{1}{\eta}\left(\mathfrak{D}(\pi_s, \pi_s^k) - \mathfrak{D}(\pi_s', \pi_s^k)\right)\right]\right| \\
&\leq \mathbb{E}_{s\sim\mu^k}\left[\left(H^2 + L/\eta\right)\left\|\pi_s - \pi_s'\right\|_1\right] \\
&= \left(H^2 + L/\eta\right)\left\|\pi - \pi'\right\|_{L^1(\mu^k),1}.
\end{aligned}$$

By a similar argument, using the good event (1):

$$\left|\widehat{\Phi}_k(\pi) - \widehat{\Phi}_k(\pi')\right| \leq \left(A\widetilde{H}H + L/\eta\right) \|\pi - \pi'\|_{L^1(\mu^k),1} .$$

Further,

$$\|\pi - \pi'\|_{L^1(\mu^k),1} \leq \|\pi - \pi'\|_{\infty,1} ,$$

hence, we have that for any $\pi \in \widetilde{\Pi}$, there exists $\pi' \in \mathrm{Net}(\varepsilon_{\mathrm{gen}}, \widetilde{\Pi})$ such that:

$$\begin{aligned}
\left|\Phi_k(\pi) - \widehat{\Phi}_k(\pi)\right| &\leq |\Phi_k(\pi) - \Phi_k(\pi')| + \left|\Phi_k(\pi') - \widehat{\Phi}_k(\pi')\right| + \left|\widehat{\Phi}_k(\pi') - \widehat{\Phi}_k(\pi)\right| \\
&\leq \frac{CAH^2D}{\sqrt{N}} + \left|\Phi_k(\pi') - \widehat{\Phi}_k(\pi')\right| + \frac{CAH^2D}{\sqrt{N}} \\
&\leq \frac{3CAH^2D}{\eta}\sqrt{\frac{\log \frac{2KN\mathcal{N}(\epsilon,\widetilde{\Pi})}{\delta}}{N}} =: \varepsilon_{\mathrm{gen}}/2.
\end{aligned}$$

Now, let $\hat{\pi}_\star^{k+1} = \arg\min_{\pi\in\widetilde{\Pi}} \widehat{\Phi}_k(\pi)$ and $\pi_\star^{k+1} = \arg\min_{\pi\in\widetilde{\Pi}} \Phi_k(\pi)$, then we have:

$$\begin{aligned}
\Phi_k(\pi^{k+1}) &\leq \widehat{\Phi}_k(\pi^{k+1}) + \varepsilon_{\mathrm{gen}}/2 \\
&\leq \widehat{\Phi}_k(\hat{\pi}_\star^{k+1}) + \varepsilon_{\mathrm{gen}}/2 + \varepsilon_{\mathrm{erm}} \\
&\leq \widehat{\Phi}_k(\pi_\star^{k+1}) + \varepsilon_{\mathrm{gen}}/2 + \varepsilon_{\mathrm{erm}} \\
&\leq \Phi_k(\pi_\star^{k+1}) + \varepsilon_{\mathrm{gen}} + \varepsilon_{\mathrm{erm}} \\
&= \min_{\pi\in\widetilde{\Pi}} \Phi_k(\pi) + \varepsilon_{\mathrm{gen}} + \varepsilon_{\mathrm{erm}}.
\end{aligned}$$

This completes the proof. □

F Experiments implementation details

In this section, we provide further details on the experimental setup where we evaluate the VGD condition parameters. The pseudocode used for the experiments is give in Algorithm 12. The environments we tested on consider rewards and not cost functions, therefore the code and discussion below should be understood as having the negative reward as the cost function (we opt to maintain the cost formulation here to better align with our original setup). Additionally, the environments considered are finite-horizon and the objective is undiscounted.

Evaluating "Sub Optimality" at iteration k. For each (seed, environment) combination, after the execution was concluded, we take the maximum value attained by the actor iterates, $\widehat{V}^\star = \min_{k\in K} \widehat{V}(\pi^k)$, where $\widehat{V}(\pi^k)$ is estimated using rollouts during experiment execution. We made sure to run the experiments for long enough (i.e., for large enough K) so that the algorithm converges. Sub optimality of iteration k is then given by $\widehat{V}(k) - \widehat{V}^\star$.

Evaluating "VGD Ratio" at iteration k; ν_k. Given the sub-optimality evaluated as described in the previous paragraph, we report ν_k by computing the following:

$$\nu_k := \frac{\widehat{V}(\pi^k) - \widehat{V}^\star}{\left\langle \widehat{\nabla} V(\pi^k), \pi^k - \tilde{\pi}^{k+1} \right\rangle}.$$

Local optima. As mentioned in Section 5, local optima was only an issue (i.e., the case that $\widehat{V}^\star \neq V^\star(\Pi_{\mathrm{actor}})$) in the MinAtar environments. We note that our analysis holds just the same under the assumption the VGD condition is satisfied for $(\nu, 0)$ w.r.t. a given target value $\widehat{V}^\star$ which is not necessarily the in-class optimal one. This may be interpreted as an execution specific value of $\varepsilon_{\mathrm{vgd}} = \widehat{V}^\star - V^\star(\Pi_{\mathrm{actor}})$. Thus, while the environments in question may not satisfy VGD globally, it seems convergence behavior may nonetheless be governed by the effective VGD parameters encountered during execution.

Algorithm 12 Pseudocode for VGD parameter evaluation

Initialize two actor neural networks $\Pi_{\text{actor}}, \Pi_{\text{vgd}}$
Initialize $\pi^1 \in \Pi_{\text{actor}}$
for $k = 1$ **to** K **do**
  // Gradient estimation phase:
  Rollout π^k to collect N environment timesteps $\mathcal{D}^k = \left\{s_i^k\right\}_{i=1}^N$
  for $s \in \mathcal{D}^k, a \in \mathcal{A}$ **do**
    Rollout n_{rep} episodes of π^k starting from s, a.
    Set $\widehat{Q}_{s,a}^k \leftarrow$ average of returns from the previous step.
  end for
  // (Note: we treat each state as if it were an independent sample from μ^k, even though it is not.)

  // Update actor:
  Train π^{k+1} for n_{epochs} epochs, with n_{mbs} mini-batches in each epoch:

$$\pi^{k+1} \approx \arg\min_{\pi \in \Pi_{\text{actor}}} \frac{1}{N} \sum_{s \in \mathcal{D}^k} \left\langle \widehat{Q}_s^k, \pi_s - \pi_s^k \right\rangle + \frac{1}{2\eta} \left\| \pi_s - \pi_s^k \right\|_2^2$$

  // Evaluate VGD:
  Initialize $\tilde{\pi}^{k+1} \leftarrow \pi^{k+1}$
  for $n_{\text{epochs}}^{\text{vgd}}$ epochs, with $n_{\text{mbs}}^{\text{vgd}}$ mini-batches in each epoch:

$$\tilde{\pi}^{k+1} \approx \arg\max_{\pi \in \Pi_{\text{vgd}}} \frac{1}{N} \sum_{s \in \mathcal{D}^k} \left\langle \widehat{Q}_s^k, \pi_s^k - \pi_s \right\rangle$$

  Estimate $\widehat{H}^k$ the average episode length of π^k
  Report $\text{GradVGD}^k = \frac{1}{N} \sum_{s \in \mathcal{D}^k} \left\langle \widehat{H}^k \widehat{Q}_s^k, \pi_s^k - \tilde{\pi}_s^{k+1} \right\rangle =: \left\langle \widehat{\nabla} V(\pi^k), \pi^k - \tilde{\pi}^{k+1} \right\rangle$
end for
return $\hat{\pi} := \pi^{K+1}$

Hyperparameters. For the VGD actor, we used the AdamW optimizer with stepsize $5e-4$, $n_{\text{epochs}}^{\text{vgd}} = 100, n_{\text{mbs}}^{\text{vgd}} = 4$ accross all experiments. Below, $N = N_{\text{envs}} \times N_{\text{steps}}$ means N_{steps} where executed in N_{envs} in parallel. For the actor optimizer we used the Adam optimizer with step size η_{opt} (this is the step size of the "inner" optimization).

- **Cartpole:** $K = 40, N = 4 \times 500, n_{\text{epochs}} = 100, n_{\text{mbs}} = 4, n_{\text{rep}} = 5, \eta_{\text{opt}} = 2e-4, \eta = 0.01$.
- **Acrobot:** $K = 100, N = 8 \times 500, n_{\text{epochs}} = 100, n_{\text{mbs}} = 4, n_{\text{rep}} = 20, \eta_{\text{opt}} = 4e-4$ with linear annealing, $\eta = 0.1$.
- **SpaceInvaders-MinAtar:** $K = 600, N = 16 \times 1000, n_{\text{epochs}} = 4, n_{\text{mbs}} = 8, n_{\text{rep}} = 5, \eta_{\text{opt}} = 5e-3$ with linear annealing, $\eta = 0.1$.
- **Breakout-MinAtar:** $K = 200, N = 16 \times 1000, n_{\text{epochs}} = 100, n_{\text{mbs}} = 8, n_{\text{rep}} = 5, \eta_{\text{opt}} = 5e-3$ with linear annealing, $\eta = 0.1$.

Additional comments.

- The architecture of both actor models is identical to that of the purejaxrl implementation [Lu et al., 2022].
- For Breakout-MinAtar and SpaceInvaders-MinAtar, execution took approximately 90-120 minutes per seed, on an NVIDIA-RTX-A5000 GPU. The experiments for Cartpole and Acrobot were run on similar hardware and took under 20 minutes for all 10 seeds.