Global Optimality for Constrained Exploration via Penalty Regularization

Florian Wolf[*†] **Ilyas Fatkhullin**[‡] **Niao He**[‡]

[†]California Institute of Technology [‡]ETH Zürich & ETH AI Center

Abstract

Efficient exploration is a central problem in reinforcement learning and is often formalized as maximizing the entropy of the state-action occupancy measure. While unconstrained maximum-entropy exploration is relatively well understood, real-world exploration is often constrained by safety, resource, or imitation requirements. This constrained setting is particularly challenging because entropy maximization lacks additive structure, rendering Bellman-equation-based methods inapplicable. Moreover, scalable approaches require policy parameterization, inducing non-convexity in both the objective and the constraints. To our knowledge, the only prior model-free policy-gradient approach for this setting under general policy parameterization is due to Ying et al. 2025 [141]. Unfortunately, their guarantees are limited to weak regret and ergodic averages, which do not imply that the final output is a single deployable policy that is near-optimal and nearly feasible. In this work we take a different approach to this problem, and propose Policy Gradient Penalty (PGP) method, a single-loop policy-space method that enforces general convex occupancy-measure constraints via quadratic-penalty regularization. PGP constructs pseudo-rewards that yield gradient estimates of the penalized objective, subsequently exploiting the classical Policy Gradient Theorem. We further establish the regularity of the penalized objective, providing the smoothness properties needed to justify the convergence of PGP. Leveraging hidden convexity and strong duality, we then establish global last-iterate convergence guarantees, attaining an ε-optimal constrained entropy value with ε-bounded constraint violation despite policy-induced non-convexity. We validate PGP through ablations on a grid-world benchmark and further demonstrate scalability on two challenging continuous-control tasks.[2]

1 Introduction

Efficient exploration is a fundamental challenge in reinforcement learning (RL), particularly in the absence of an external reward signal, where the goal is to acquire broad and informative coverage of the environment rather than optimize task-specific returns. A principled line of work formulates pure exploration as maximizing the entropy of the state-action occupancy measure induced by a policy, encouraging diverse visitation of the environment [51]. This viewpoint has led to substantial progress in entropy-based exploration, with both theoretical and practical advances [93, 115].

In realistic systems, however, exploration is inherently constrained not only by safety limits and resource budgets, but crucially by imitation requirements, where policies must remain close to a given prior or expert behavior while still exploring novel regions of the state-action space [42]. Despite

[*]Corresponding author: fwolf@caltech.edu

[2]Code available under: https://github.com/Flo-Wo/penalty-based-constrained-exploration.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

their importance, existing approaches to constrained entropy-based exploration are largely heuristic [134, 116]. The model-based approaches of Agarwal et al. [2], Bai et al. [9] operate with explicit representations of the state-action occupancy measure, which limits their applicability in large-scale or continuous settings where function approximation is required. In contrast, Ying et al. [141] provide only ergodic and weak-regret guarantees under restrictive assumptions.

A natural way to model safety, cost, and imitation requirements is to express them as convex constraints on the state-action occupancy measure, yielding a unified and expressive formulation for both finite- and infinite-horizon Markov decision processes. Achieving scalability, however, necessitates policy parameterization, which obscures this convex structure and eliminates the additive Bellman structure underlying standard dynamic-programming and actor-critic methods. Together with enforcing constraints under stochastic, truncated trajectory sampling with biased gradients, this leads to highly challenging optimization problems.

Contributions. We develop a penalty framework for constrained maximum-entropy exploration that operates directly in policy space while retaining the hidden-convex structure of occupancy-measure optimization.[3] Our contributions are:

- (**C1**) We introduce PGP, a *single-loop, primal-only* Policy Gradient Penalty method for constrained maximum-entropy exploration that operates directly in policy space and enforces non-convex constraints via a quadratic penalty. By explicitly utilizing the RL structure of the problem, we construct *pseudo-rewards*, enabling single batch penalty gradient estimation.
- (**C2**) We prove *global, non-asymptotic last-iterate convergence* of PGP under biased stochastic gradients, despite the non-convexity induced by policy parametrization. The analysis quantifies the smoothness of the penalty formulation, a key property needed to establish global convergence, and provides a principled choice of the penalty regularization parameter.
- (**C3**) In the finite-MDP setting covered by our theory, we validate PGP on a grid world, demonstrating robustness to penalty tuning and gradient noise and providing practical guidance for choosing the penalty parameter β. Beyond our theory, continuous state-action experiments demonstrate scalability and stable late-iterate feasibility.

2 Related Work

Entropy Maximization. Maximizing the entropy of the state-action occupancy measure was introduced by Hazan et al. [51] as a principled formulation of reward-free exploration. Subsequent work has focused on improving computational efficiency and practical performance, including non-parametric entropy estimation [92, 93], representation learning [111, 138], and successor-based approaches [76, 77]. On the theoretical side, Tiapkin et al. [115] established fast convergence rates for maximum-entropy exploration, while Zamboni et al. [144, 145] extended entropy-based exploration to partially observable Markov decision processes (MDPs). These works focus almost exclusively on the unconstrained setting. While they exploit the convex structure of the entropy objective in the occupancy-measure space, they do not address constraints and therefore cannot guarantee feasibility in safety- or resource-constrained environments.

Constrained Entropy-Based Exploration. Several recent works have studied constrained entropy maximization empirically. Yang and Spaan [134] propose constrained entropy maximization for safe, task-agnostic exploration, and Tiboni et al. [116] apply entropy maximization under performance chance constraints to domain randomization for sim-to-real transfer in robotics. Related ideas appear in transfer and imitation settings [16, 135, 60], observer gain tuning [61, 81] and system identification under performance constraints [126, 155]. However, these methods are heuristic in nature and do not provide (global) convergence guarantees, even in tabular settings.

More broadly, constrained and safe RL has been extensively studied under the Constrained Markov decision process (CMDP) formulation [4], for instance, using primal-dual methods [38, 43, 139, 78, 123, 49, 37, 10, 39, 71, 87, 40], primal-only approaches [1, 25, 28, 136, 131, 120, 100, 22], and in Bayesian settings [5, 124, 6].

[3] In Appendix D, we further extend our framework beyond RL to constrained hidden-convex optimization problems with possibly non-smooth objectives and constraints, which could be of independent interest.

Convex Reinforcement Learning. Unconstrained convex, or general-utility, RL generalizes classical RL by allowing the objective to be a convex functional of the state-action occupancy measure, instead of standard additive rewards [143]. In the unconstrained setting, Zhang et al. [150] establish global convergence guarantees in tabular MDPs, with subsequent extensions to function approximation and variance reduction [152, 11, 110], as well as to large-scale problems [52, 12]. Finite-trial formulations [94] have also been studied [95, 96]. Related directions include multi-agent convex RL [140], a mean-field perspective [48], and extensions to risk-sensitive MDPs [128], submodular objectives [104, 35], and robust convex RL [24]. Online variants of unconstrained convex RL have been investigated in [89, 83, 90, 13]. Closely related in spirit, Kalogiannis et al. [59] study policy gradient convergence in imperfect-information extensive-form games. In the optimization literature, Fatkhullin et al. [45] analyze algorithms for unconstrained stochastic optimization problems under hidden convexity and Fatkhullin et al. [46] develop algorithms for hidden convex constrained problems. We refer to Appendix A for additional related work and a detailed technical comparison of our contributions to the two most closely related works, Zhang et al. [150] and Fatkhullin et al. [45].

In contrast, significantly fewer works address convex RL under complex (e.g., safety or imitation) constraints. Agarwal et al. [2] study constrained convex MDPs (CCMDPs) using a model-based approach over the occupancy-measure space with count-based approximations and direct policy parametrization, relying on a strong form of Slater's condition. Bai et al. [9] relax this requirement to the classical Slater condition and propose a Kullback Leibler (KL) regularized primal-dual algorithm, still operating in a model-based occupancy-measure formulation. More recently, [141] is the first work to study CCMDPs under a general softmax policy parametrization using a policy-gradient approach in policy parameter space. However, the obtained guarantees are limited to weak regret and ergodic averages, i.e., $\frac{1}{N}\sum_{k=1}^{N} F_1(\theta^{(k)}) - F^\star \leq \varepsilon$ and $\left[\frac{1}{N}\sum_{k=1}^{N} F_2(\theta^{(k)})\right]_+ \leq \varepsilon$, see for example the discussion in [91]. Such guarantees permit cancellations in constraint violations and, moreover, are not operational in practice: averaging or sampling from iterates in parameter space is generally ill-defined for policy networks. Relatedly, Zhang et al. [148] study convex constrained MDPs in an offline, model-based setting using primal-dual methods under different assumptions. Overall, existing approaches to constrained convex RL suffer from at least one of the following limitations: they rely on model-based or count-based formulations that do not scale beyond small discrete state-action spaces; they require (forms of) Slater's condition which might fail to hold in practice; and they provide only ergodic guarantees, precluding *at-any-time* convergence of the last iterate.

More generally, it was observed for constrained MDPs [131, 55] that primal-only methods can be preferable in practice, due to fewer hyperparameters, simpler implementations, and faster convergence resulting from the absence of delayed dual updates.

3 Preliminaries

Notation. For a finite set A, we denote with $|A|$ the cardinality and with $\Delta(A)$ the $(|A|-1)$-dimensional simplex, i.e. the space of probability distributions over A. If not stated differently, $\|\cdot\|$ denotes the standard ℓ^2-norm for vectors and the spectral norm for matrices, and $\langle\,\cdot\,,\,\cdot\,\rangle$ is the Euclidean inner product. A differentiable function $F : \Theta \to \mathbb{R}$ is L-smooth if the gradient is L-Lipschitz continuous, $L \in \mathbb{R}_{\geq 0}$. For an integer $k \in \mathbb{N}$ define the set $[k] \coloneqq \{1, \dots, k\}$. A notation $\mathcal{O}(\,\cdot\,)$ captures all dependencies except for the accuracy ε and $\widetilde{\mathcal{O}}(\,\cdot\,)$ hides logarithmic terms in $1/\varepsilon$.

Markov Decision Process (MDP). A discrete-time infinite-horizon discounted Markov Decision Process is a tuple $\mathbb{M} \coloneqq (\mathcal{S}, \mathcal{A}, r, \mathbb{P}, \gamma, \mu_0)$, where $\mathcal{S}$ and $\mathcal{A}$ are finite sets of states and actions, respectively. At a discrete timestep $t \in \mathbb{N}_0$, being in state $s_t \in \mathcal{S}$ taking an action $a_t \in \mathcal{A}$ yields a reward $r_t = r(s_t, a_t)$, according to the reward function $r : \mathcal{S} \times \mathcal{A} \to [-r_{\max}, r_{\max}]$, and a new state $s_{t+1} \sim \mathbb{P}(\,\cdot\,|s_t, a_t)$ according to the transition probability kernel $\mathbb{P} : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$. $r_{\max} > 0$ gives an upper and lower bound on the reward. We define the *value function* of a trajectory $\tau \coloneqq (s_t, a_t)_{t \in \mathbb{N}_0}$ as the cumulative discounted reward $V(\tau) \coloneqq (1-\gamma) \cdot \sum_{t=0}^{\infty} \gamma^t r(s_t, a_t)$, where μ_0 denotes the initial state distribution, i.e. $s_0 \sim \mu_0$, and $\gamma \in (0, 1)$ is the so-called discount factor. A mapping $\pi : \mathcal{S} \to \Delta(\mathcal{A})$ is called a (stationary) *policy*, and Π denotes set of all stationary policies. Combined with the initial distribution μ_0, it induces a probability measure over finite-length trajectories $\tau = (s_t, a_t)_{t=0}^{T-1}$ by sampling $a_t \sim \pi(\,\cdot\,|s_t)$ at each timestep $t = 0, \dots, T$, $T \in \mathbb{N}$, i.e. $p(\tau) \coloneqq \mu_0(s_0)\pi(a_0|s_0) \cdot \prod_{t=1}^{T-1} \mathbb{P}(s_t|s_{t-1}, a_{t-1})\pi(a_t|s_t)$.

Policy Parametrization. Throughout this work, we will assume the common softmax policy parameterization

$$\pi_\theta(a|s) \coloneqq \frac{\exp(\psi(s,a;\theta))}{\sum_{a' \in \mathcal{A}} \exp(\psi(s,a';\theta))},$$

where $\theta \in \Theta \subset \mathbb{R}^p$, $s \in \mathcal{S}$, $a \in \mathcal{A}$, and a smooth function $\psi : \mathcal{S} \times \mathcal{A} \times \Theta \to \mathbb{R}$.

State-action Occupancy Measure (OM). For any policy π_θ, we define the (state-action) occupancy measure $\lambda^{\pi_\theta} : \mathcal{S} \times \mathcal{A} \to [0,1]$ as

$$\lambda^{\pi_\theta}(s,a) \coloneqq (1-\gamma) \sum_{t=0}^{\infty} \gamma^t \mathbb{P}_{\mu_0,\pi_\theta}(s_t = s, a_t = a),$$

where $\mathbb{P}_{\mu_0,\pi_\theta}$ denotes the probability distribution of the Markov chain $(s_t, a_t)_{t \in \mathbb{N}}$ induced by the policy π_θ and the initial state distribution μ_0 under the transition probability kernel $\mathbb{P}$. We denote by $\Lambda \coloneqq \{\lambda^\pi \ : \ \pi \in \Pi\}$ the set of all state-action occupancy measures induced by the stationary policies Π. Since $|\mathcal{S}|, |\mathcal{A}| < \infty$, we will interpret a $\lambda \in \Lambda$ as an element of $[0,1]^{|\mathcal{S}| \times |\mathcal{A}|}$. For a parametrized policy, we will use the notations $\lambda(\theta)$ and λ^{π_θ} interchangeably. Intuitively, since the OM captures the state-action visitation frequencies induced by a policy, it provides a principled foundation for defining exploration as maximizing the entropy of the induced state-action distribution.

4 Problem Formulation

For a convex, L-smooth constraint $\mathcal{R} : \Lambda \to \mathbb{R}$ formulated in the state-action-occupancy measure, we formulate the following constrained-max-entropy problem:

$$\max_{\theta \in \Theta} F_1(\theta) \coloneqq \mathcal{H}(\lambda^{\pi_\theta}) = -\mathbb{E}_{(s,a)\sim\lambda^{\pi_\theta}}[\log \lambda^{\pi_\theta}(s,a)] \quad \text{s.t.} \quad F_2(\theta) \coloneqq \mathcal{R}(\lambda^{\pi_\theta}) \leq 0. \tag{CME}$$

We assume only that there exists a stationary policy $\pi_{\text{feas}} \in \Pi$ satisfying $\mathcal{R}(\lambda^{\pi_{\text{feas}}}) \leq 0$. Since Λ is a compact polytope in a finite MDP, and $\mathcal{H}$ and $\mathcal{R}$ are continuous, the corresponding problem over all stationary policies admits an optimizer $\pi^* \in \Pi$. We denote its value by $\mathcal{H}^* \coloneqq \mathcal{H}(\lambda^{\pi^*})$ and set $F^* \coloneqq \mathcal{H}^*$. We call $\hat{\theta} \in \Theta$ an $(\varepsilon,\varepsilon)$-optimal solution if $|\mathbb{E}\Big[F_1(\hat{\theta}) - F^*\Big]| \leq \varepsilon$ and $\mathbb{E}\Big[[F_2(\hat{\theta})]_+\Big] \leq \varepsilon$, where $\varepsilon > 0$. At first sight, optimizing (CME) over $\theta \in \Theta$, rather than directly over all stationary policies, may appear restrictive. The following proposition shows that the (anchored) tabular softmax family is dense in the set of stationary policies. Theorem C.2 (Appendix) then shows that this density is sufficient for the stated approximate objective and feasibility guarantees.

Proposition 4.1 (Anchored Softmax Approximation of Stationary Policies)**.** *Let $\mathbb{M}$ be a finite MDP with an initial distribution $\min_{s\in\mathcal{S}} \mu_0(s) > 0$, and fix a reference action $a_0 \in \mathcal{A}$. Consider the anchored tabular logits $\psi(s,a_0;\theta) = 0$ and $\psi(s,a;\theta) = \theta_{s,a}$ for $a \neq a_0$, with parameter space $\Theta_R = [-R,R]^{\mathcal{S}\times(\mathcal{A}\setminus\{a_0\})}$. For every stationary policy $\pi^* \in \Pi$, including deterministic ones, and every $\varepsilon \in (0,1]$, there exist $R = \mathcal{O}(\log(|\mathcal{A}|/\varepsilon))$ and a parameter $\theta \in \Theta_R$ such that*

$$\max_{s\in\mathcal{S}} \|\pi_\theta(\cdot|s) - \pi^*(\cdot|s)\|_1 \leq \varepsilon,$$

i.e. the family of anchored tabular softmax classes $\bigcup_{R>0}\{\pi_\theta \ : \ \theta \in \Theta_R\}$ is dense in the set of stationary policies.

Proof. Full details in Appendix C, we show that fixing the reference logit makes the parameter-to-occupancy map invertible with a Lipschitz inverse on Θ_R. □

The problem formulation (CME) includes, but is not limited to, the following **examples**:

- **(Cumulative Cost)** A safety constraint can be given by the discounted cumulative cost $F_2(\theta) \coloneqq (1-\gamma)\mathbb{E}_{\mu_0,\pi_\theta}[\sum_{t=0}^{\infty} \gamma^t c(s_t,a_t)] - c_{\max}$, for a cost vector $c \in \mathbb{R}^{\mathcal{S}\times\mathcal{A}}$. In this case (CME) corresponds to finding the policy with the maximum entropy among all the policies solving the environment $\mathcal{R}(\lambda^{\pi_\theta}) \coloneqq \langle c, \lambda^{\pi_\theta}\rangle - c_{\max}$.

- **(Apprenticeship Learning)** Closeness to an expert's or prior trajectory can be encoded using the KL divergence $\mathcal{R}(\lambda^{\pi_\theta}) \coloneqq \mathsf{D}_{\mathsf{KL}}\left(\lambda^{\pi_\theta} \| \lambda^{\pi_{\text{ref}}}\right) - \mathcal{R}_{\max}$, with respect to the state-action-occupancy measure of a reference policy $\lambda^{\pi_{\text{ref}}}$.

Naturally, our framework can be extended to multiple constraints, by considering a vector-valued $\mathcal{R}$ and dealing with the Jacobian instead of the gradient.

5 Policy Gradient Penalty Method

Our approach proceeds by transforming (CME) into an unconstrained optimization problem through a quadratic penalty reformulation:

$$\min_{\theta \in \Theta} P(\theta) \coloneqq -F_1(\theta) + \beta \mathcal{L}(F_2(\theta)), \tag{PEN}$$

where $\mathcal{L} \coloneqq [\,\cdot\,]_+^2$. In Appendix D, we show that for non-smooth constrained hidden convex problems the use of the exact penalty function $\mathcal{L}(\,\cdot\,) \coloneqq [\,\cdot\,]_+$ is favorable for the theoretical analysis. Additionally, for $\theta \in \Theta$ we define the quadratic-penalty *optimality value gap* as $P^{\mathsf{opt}}_{2,\beta}(\theta) \coloneqq P(\theta) + F^*$.

(Stochastic) Policy Gradient. To derive the gradient of a general smooth, convex $F : \Lambda \to \mathbb{R}$ with respect to $\theta \in \Theta$ we can use the policy gradient theorem [113] and the chain rule. With $V^{\pi_\theta}(r) = \langle \lambda(\theta), r \rangle$ we have

$$[\nabla_\theta \lambda(\theta)]^\top r = \nabla_\theta V^{\pi_\theta}(r) = (1-\gamma)\mathbb{E}_{\mu_0,\pi_\theta}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t,a_t) \sum_{k=0}^{t} \nabla_\theta \log(\pi_\theta(a_k|s_k))\right] \tag{PG-Thm}$$

by denoting with $\nabla_\theta \lambda(\theta)$ the Jacobian of the (vector-valued) mapping $\lambda(\theta)$, we can apply the chain rule and obtain

$$\nabla_\theta F(\lambda(\theta)) = [\nabla_\theta \lambda(\theta)]^\top \cdot \nabla_\lambda F(\lambda(\theta)) = \nabla_\theta V^{\pi_\theta}(r)\Big|_{r=\nabla_\lambda F(\lambda(\theta))},$$

i.e. we can compute the policy gradient of the objective and the constraint estimating the state-action occupancy measure and then applying the standard REINFORCE [151] estimator using H-long trajectory truncations

$$\nabla_\theta V^{\pi_\theta}(r) \approx (1-\gamma) \sum_{t=0}^{H-1} \left(\sum_{k=t}^{H-1} \gamma^k r(s_k,a_k)\right) \nabla_\theta \log(\pi_\theta(a_t|s_t)) =: g_H(\theta,\tau,r)$$

The OM can be estimated using Monte-Carlo rollouts, i.e. for a batch of $(\tau_b)_{b\in B}$, $|B| \geq 1$ i.i.d. trajectories, we define

$$\widehat{\lambda}\left((\tau_b)_{b\in B}\right) \coloneqq \frac{1-\gamma}{|B|} \sum_{b\in B} \sum_{t=0}^{H-1} \gamma^t \delta_{s_t^{(b)}, a_t^{(b)}}, \tag{λ_H^{MC}-Est}$$

where for every $(s,a) \in \mathcal{S} \times \mathcal{A}$ the vector $\delta_{s,a} \in \{0,1\}^{\mathcal{S}\times\mathcal{A}}$ has a one at the entry (s,a) and zeros elsewhere, yielding an unbiased estimate of the *truncated* occupancy measure

$$\lambda_H^{\pi_\theta}(s,a) \coloneqq (1-\gamma) \sum_{t=0}^{H-1} \gamma^t \mathbb{P}_{\mu_0,\pi_\theta}(s_t = s, a_t = a).$$

In the case of continuous state-action spaces, we are using function approximation and a maximum likelihood estimator (λ_H^{MLE}-Est), with implementation details in Appendix E.3.1. In total, we define the (biased) estimate

$$\hat{g}_{H,B}(\nabla_\lambda F, \theta, (\tau_b)_{b\in[B]}) \coloneqq \frac{1}{B} \sum_{b=1}^{B} g_H\left(\theta, \tau_b, r = \nabla_\lambda F(\lambda)\big|_{\lambda=\widehat{\lambda}((\tau_i)_{i\in[B]})}\right) \tag{∇_θ-Est}$$

of the truncated policy-gradient $\nabla_\theta F(\lambda_H(\theta))$. We present in Appendix B, how this translates to approximation guarantees with respect to the true, infinite horizon gradient.

Pseudo-rewards and single-trajectory-batch penalty gradients. A key algorithmic novelty lies in the construction of *pseudo-rewards* $g_\diamond^{(k)}$, $\diamond \in \{O, C\}$, introduced in Line 5. In generic stochastic optimization, both, exact and quadratic penalty formulations require applying the (generalized) chain rule through the constraint function, which canonically necessitates separate stochastic estimators for the function value and its (sub-)gradient. This mismatch induces bias and has led prior work on stochastic penalty methods and ALM to either restrict attention to deterministic constraints [63, 121, 122], incur additional sample complexity via multiple gradient or function evaluations per iteration [27, 79, 133] or rely on momentum and variance-tracking schemes to control the resulting bias [74, 69].

Algorithm 1 POLICY GRADIENT PENALTY METHOD

1: **Input:** Objective $\mathcal{H}$, constraint $\mathcal{R}$, initial policy $\theta^{(0)} \in \Theta$, number of iterations $N \in \mathbb{N}$, batch size $B \in \mathbb{N}$, step size $\eta > 0$, penalty parameter $\beta > 0$
2: **for** $k = 0$ **to** N **do**
3: Sample trajectory batch $\tau_b^{(k)} \overset{\text{i.i.d.}}{\sim} \pi_{\theta^{(k)}}$, $b \in [B]$
4: Estimate occupancy measure $\lambda^{(k)} := \widehat{\lambda}((\tau_b^{(k)})_{b\in[B]})$ via (λ_H^{MC}-Est) or (λ_H^{MLE}-Est)
5: Compute pseudo-rewards via Backpropagation

$$g_O^{(k)} := \nabla_\lambda(-\mathcal{H}(\lambda))\big|_{\lambda=\lambda^{(k)}},$$
$$g_C^{(k)} := \nabla_\lambda(\mathcal{L} \circ \mathcal{R}(\lambda))\big|_{\lambda=\lambda^{(k)}}$$

6: Gradient estimate via (∇_θ-Est) for $\diamond \in \{O, C\}$:

$$\hat{g}_\diamond^{(k)} := \hat{g}_{H,B}(g_\diamond^{(k)}, \theta^{(k)}, \{\tau_b^{(k)}\}_{b\in[B]}),$$

7: Stochastic Gradient Descent Step

$$\theta^{(k+1)} \leftarrow \text{proj}_\Theta\left[\theta^{(k)} - \eta \cdot (\hat{g}_O^{(k)} + \beta \hat{g}_C^{(k)})\right]$$

8: **end for**
9: **Return:** Last policy $\theta^{(N+1)}$.

Our method fundamentally departs from this paradigm by explicitly exploiting the RL structure of the problem. Leveraging the Policy Gradient Theorem (PG-Thm), we reinterpret gradients of both the objective and the penalty term as policy gradients with respect to constructed pseudo-rewards. This yields the estimator in (∇_θ-Est), which allows us to compute a stochastic gradient of the penalized objective using a single trajectory batch, without auxiliary function-value estimation.

Importantly, the occupancy measure is estimated only once from the full batch, and the resulting common pseudo-reward is reused on every trajectory in that same batch. In particular, the quadratic-penalty gradient contains $2\beta[\mathcal{R}(\lambda)]_+\nabla\mathcal{R}(\lambda)$, whose two factors both depend on the unknown occupancy measure. Using the same trajectory batch to estimate these factors and form the policy-gradient estimate therefore yields a dependent and biased estimator. We show in Lemma B.2 and Corollary B.5 that our pseudo-reward construction controls both truncation and occupancy-estimation bias, as well as the estimator variance, without requiring a model, an exact value function, nested sampling, or auxiliary function evaluations.

5.1 Theoretical Analysis

The key theoretical insight is the hidden convex structure induced by the occupancy-measure formulation. By mapping a policy π to its corresponding occupancy measure, the penalized problem becomes convex and admits a one-step contraction inequality under biased stochastic gradients. Translating this contraction back to policy space allows us to unroll the recursion over iterations, yielding global last-iterate convergence guarantees. We make the following (standard) assumptions:

Assumption 5.1 (Softmax-Parametrization)**.** The function $\psi : \mathcal{S} \times \mathcal{A} \times \Theta \to \mathbb{R}$ satisfies $\psi \in C^2(\mathcal{S}, \mathcal{A}, \Theta; \mathbb{R})$, and there exist constants $\ell_\psi, L_\psi > 0$ such that (i) $\max_{s\in\mathcal{S},a\in\mathcal{A}} \sup_{\theta\in\Theta} \|\nabla_\theta \psi(s, a, \theta)\| \le \ell_\psi$ and (ii) $\max_{s\in\mathcal{S},a\in\mathcal{A}} \sup_{\theta\in\Theta} \|\nabla_\theta^2 \psi(s, a, \theta)\| \le L_\psi$.

Under these assumptions, we show the regularity of the *score function* $\nabla_\theta \log \pi_\theta(\,\cdot\,)$ in Appendix B.

Assumption 5.2 (Occupancy-Measure Parametrization)**.** The parameter-to-occupancy mapping $\lambda : \Theta \to \Lambda$, $\theta \mapsto \lambda^{\pi_\theta}$, satisfies:

- (Locally Invertible) For every parameter $\theta \in \Theta$ there exists a neighborhood $\theta \in \mathcal{U}_\theta \subset \Theta$ such that $\lambda^{\pi_\theta}|_{\mathcal{U}_\theta}$ is a bijection between $\mathcal{U}_\theta$ and $\mathcal{V}_\theta := \lambda^{\pi_\theta}(\mathcal{U}_\theta)$. A (local) inverse is $\lambda_{\mathcal{V}_\theta}^{-1} : \mathcal{V}_\theta \to \mathcal{U}_\theta$.
- (Lipschitz Inverse) The inverse $\lambda_{\mathcal{V}_\theta}^{-1}$ is μ_λ-Lipschitz continuous over Θ.
- (Locally Hidden Convex) There exists an $\varepsilon_\lambda > 0$ such that for all $\varepsilon \in (0, \varepsilon_\lambda]$ and all $\theta \in \Theta$ the convex combination satisfies $(1 - \varepsilon)\lambda^{\pi_\theta} + \varepsilon\lambda^{\pi^\star} \in \mathcal{V}_\theta$.

The above assumption is common and was verified in a tabular setting, e.g., [150, Prop. H.1]. For the (anchored) tabular softmax parametrization above, we verify Assumption 5.1 and Assumption 5.2 directly in Appendix C. Note that since Λ is a polytope ($\Lambda \subset [0,1]^{\mathcal{S}\times\mathcal{A}}$), its domain is bounded, i.e., there exists $\mathcal{D}_\Lambda > 0$ such that $\|\lambda(\theta) - \lambda(\theta')\| \leq \mathcal{D}_\Lambda$, for any $\theta, \theta' \in \Theta$. Due to the (local) Lipschitzness of the inverse we can also uniformly bound the domain Θ by a diameter $\mathcal{D}_\Theta$, cf. Thm. 3.2 in Zhang et al. [150]

Assumption 5.3 (L-smoothness)**.** Assume that for $\mathcal{H}$ and $\mathcal{R}$ there exist $\ell_\lambda, L_\lambda, L_{\lambda,\infty} > 0$ such that $G \in \{-\mathcal{H}, \mathcal{R}\}$ satisfies for all $\lambda, \lambda' \in \Lambda$: (i) $\|\nabla_\lambda G(\lambda)\|_\infty \leq \ell_\lambda$, (ii) $\|\nabla_\lambda G(\lambda) - \nabla_\lambda G(\lambda')\|_\infty \leq L_\lambda \|\lambda - \lambda'\|_2$, and (iii) $\|\nabla_\lambda G(\lambda) - \nabla_\lambda G(\lambda')\|_\infty \leq L_{\lambda,\infty} \|\lambda - \lambda'\|_1$.

We want to emphasize that the aforementioned assumptions are standard in convex (constrained) RL, cf. [51, 152, 11, 141, 12]. Strictly spoken, the entropy objective is only locally smooth [51, 152] and in Appendix C.2 we consider the smoothed alternative $\mathcal{H}_\sigma(\lambda) := -\sum_{s,a} \lambda(s,a) \log\big(\lambda(s,a)+\sigma\big)$, for $\sigma > 0$, explictly deriving the constants above as well as approximation guarantees and strong-duality translation back to the original entropy.

The following main theorem gives a guarantee on the optimality value gap of the penalty function, without the assumption of strong duality.

Theorem 5.4 (Biased SGD for (PEN))**.** *Let $\varepsilon > 0$ be the target accuracy. Under Assumptions 5.1, 5.2, 5.3, running Algorithm 1 with a step size $\eta = 1/L_{P,\theta}$, a batch size of $B = \widetilde{\mathcal{O}}\Big(\frac{1}{\varepsilon^2} + \frac{\beta^2}{\varepsilon^2}\Big)$ and a trajectory length of $H = \mathcal{O}\Big(\log\Big(\frac{1}{\varepsilon} + \frac{\beta}{\varepsilon}\Big)\Big)$, we obtain*

$$\mathbb{E}\Big[P^{\mathsf{opt}}_{2,\beta}(\theta^{(N)})\Big] \leq \varepsilon \quad \textit{after} \quad N \geq \mathcal{O}\left(\left(\frac{1}{\varepsilon} + \frac{\beta}{\varepsilon}\right) \cdot \log\left(\frac{P^{\mathsf{opt}}_{2,\beta}(\theta^{(0)})}{\varepsilon}\right)\right) = \widetilde{\mathcal{O}}\left(\frac{1}{\varepsilon} + \frac{\beta}{\varepsilon}\right).$$

iterations, where $P^{\mathsf{opt}}_{2,\beta}(\theta^{(0)})$ is the initial value gap and $L_{P,\theta}$ is defined in Corollary B.4.

Proof. **(Sketch)** First, our proof establishes the regularity of the penalty objective P in Lemma B.3 using Assumption 5.3. Next, we analyze Algorithm 1 by controlling the bias and variance of the SGD step, which yields global convergence. A detailed proof is available in Appendix B.2.1. □

The above theorem implies that for any penalty regularization parameter β, Algorithm 1 solves our penalty formulation (PEN) despite the bias of the stochastic gradients occurring from the estimation of the state occupancy measure in (λ^{MC}_H-Est) and propagating to (∇_θ-Est). However, the sample complexity $N \cdot B$ crucially depends on the parameter β, coming from the estimate of the smoothness constant $L_{P,\theta}$. In what follows, we will translate the result to convergence to an $(\varepsilon, \varepsilon)$-optimal solution and give a recommendation on the parameter β trading the feasibility and sample complexity. We use the following standard assumption.

Assumption 5.5 (Strong Duality)**.** We assume *strong duality* holds for (CME), i.e. for θ^* there exists a Lagrangian multiplier $\nu^* \geq 0$ such that $d^* = d(\nu^*) = -F^*$, where the *(Lagrangian) dual function* is defined as the infimum of the *Lagrangian* with respect to the primal variable θ, that is, $d(\nu) := \inf_{\theta\in\Theta} L(\theta, \nu) = \inf_{\theta\in\Theta}\big(-F_1(\theta) + \nu F_2(\theta)\big)$, and the corresponding *dual problem* as $d^* := \sup_{\nu\geq 0} d(\nu) = \sup_{\nu\geq 0} \inf_{\theta\in\Theta} L(\theta, \nu)$.

It is well known that, in general for any $\nu \geq 0$, we have $d(\nu) \leq -F^*$, i.e. weak duality always holds [99]. Assumption 5.5 is not required for convergence to the penalty objective in Theorem 5.4; it is used only for the subsequent translation to objective and feasibility guarantees through a finite multiplier. Most importantly, it is strictly weaker than Slater's condition: Slater is a sufficient certificate for strong duality [99], but is not necessary. Consequently, strong duality permits boundary-only feasible sets and equality constraints for which Slater fails [46]. Under linear safety or performance constraints, a feasible point is already sufficient, cf. Bertsekas [15, Prop. 5.3.1].

5.2 Sample Complexity Analysis

Under strong duality, we can now translate the results of Theorem 5.4 into guarantees for the optimality value gap and the constraint violation.

Corollary 5.6. *Let $\varepsilon > 0$ be the target accuracy. Under Assumptions 5.1, 5.2, 5.3 and strong duality, Assumption 5.5, running Algorithm 1 with a step size $\eta = \mathcal{O}(\varepsilon)$, a batch size of $B = \mathcal{O}\big(\varepsilon^{-4}\big)$, a penalty parameter $\beta = \mathcal{O}\big(\varepsilon^{-1}\big)$, and a trajectory length of $H = \mathcal{O}\big(\log \frac{1}{\varepsilon}\big)$, we obtain*

$$-\varepsilon \leq \mathbb{E}\Big[F_1(\theta^{(N)}) - F^*\Big] \leq \varepsilon \quad \textit{and} \quad \mathbb{E}\Big[[F_2(\theta^{(N)})]_+\Big] \leq \varepsilon$$

after $N = \widetilde{\mathcal{O}}\big(\varepsilon^{-2}\big)$ iterations, resulting in a total of $\widetilde{\mathcal{O}}\big(\varepsilon^{-6}\big)$ trajectory rollouts of length $\mathcal{O}\big(\log \frac{1}{\varepsilon}\big)$.[4]

Proof. **(Sketch)** Translating the guarantees of Theorem 5.4 under strong duality requires $\beta(\varepsilon) := (\nu^*+1)\frac{\nu^*+\sqrt{\nu^{*2}+2}}{\varepsilon} = \mathcal{O}\big(\varepsilon^{-1}\big)$, using Corollary D.22. See Appendix B.2.2 for the full derivation. □

In Appendix C.2 we show that working with the smoothed entropy with $\sigma = \frac{\varepsilon}{4|\mathcal{S}||\mathcal{A}|}$ requires only the original strong-duality assumption and preserves the $\widetilde{\mathcal{O}}\big(\varepsilon^{-6}\big)$ rollout rate for fixed dimension. Writing $d = |\mathcal{S}|\,|\mathcal{A}|$, tracking dimension explicitly yields the conservative dependence $\widetilde{\mathcal{O}}\big((d+1)^3\varepsilon^{-6}\big)$. To the best of our knowledge, this is the first $\mathrm{poly}(1/\varepsilon)$ sample complexity guarantee for constrained maximum-entropy exploration under general policy parametrization. Crucially, the guarantee is for the *last iterate* of a single-loop algorithm and applies simultaneously to the objective and constraint in (CME). The choice $\beta = \mathcal{O}\big(\varepsilon^{-1}\big)$ follows from the strong-duality translation, reflecting the scaling between feasibility enforcement and the dual multiplier. A limitation of the current analysis is the large batch size $B = \mathcal{O}\big(\varepsilon^{-4}\big)$, required to control truncation bias and occupancy-measure estimation. We expect this can be reduced, for example via variance reduction techniques [152, 11, 44]; indeed, the ablations in the next section suggest the ε^{-4} dependence is likely conservative in practice. A sharper theoretical analysis is left for future work.

6 Numerical Experiments

We validate our method in a *FrozenLake* grid-world environments which is visualized in Figure 5a, where the experiment involves maximizing the entropy while fulfilling a (linear) safety constraint. The full details and additional experiments with a non-linear norm constraint are deferred to Appendix E. All training curves report the mean and one standard deviation across seeds; the grid-world experiments use 10 seeds and the continuous experiments use 5 seeds each.

6.1 Theoretical Setting: Grid World

We compare our approach to the primal-dual (PDPG) algorithm of Ying et al. [141] for which we performed a hyperparameter search (details in Appendix E.1) and display their best-performing model in Figure 1. While our PGP-method achieves an objective function slightly below the unconstrained maximum entropy policy and meets the safety constraints, the primal-dual version exhibits weak regret in practice: the iterates oscillate around the constraint, cf. [91, Sec. 6]. This distinction persists throughout late training: over the final 1000 epochs, highlighted by the pink region ① in Figure 1, Table 2a shows that PGP satisfies both feasibility tolerances substantially more often than PDPG. Figure 2 shows that the proposed penalty formulation is highly robust to the choice of β. Across several orders of magnitude, the achieved entropy remains essentially unchanged once feasibility is reached, while constraint violations decrease monotonically. This indicates that effective constraint enforcement does not require fine-tuning of the penalty parameter and does not trade off against the exploration objective. The checkpoint analysis in Table 1a also quantifies the corresponding cost: once β is sufficiently large for feasibility, further overestimation primarily slows convergence rather than changing the final entropy or constraint value.

6.2 Beyond Theory: Continuous State-Action Space

These experiments assess practical scalability and whether the robustness to gradient noise and the choice of β observed in the finite setting transfers to continuous state-action spaces. Extending our theoretical analysis to this setting is an important direction for future work. The main challenge in the continuous setting lies in estimating approximating the state-action occupancy measure. We address this using the current state-of-the-art maximum-likelihood estimator of Barakat et al. [12], with technical details deferred to Appendix E.3.

[4] The expectation is with respect to the randomness of the MDP, the policy, and the initial state distribution.

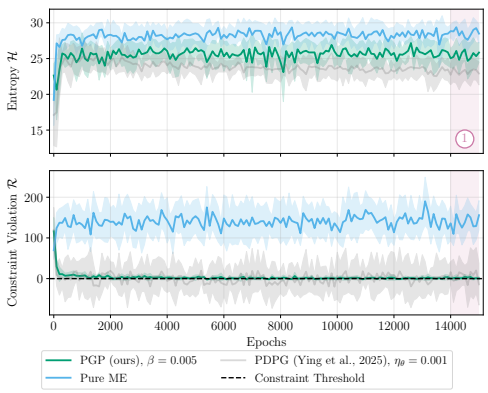


Figure 1: **Solving constrained maximum-entropy exploration** using Policy Gradient Penalty Method (PGP), unconstrained Maximum-Entropy (ME) exploration, and the primal-dual (PDPG) method of [141]. Area ① marks the final 1k epochs evaluated in Table 2a. Policies are visualized in Figure 9 (Appendix).

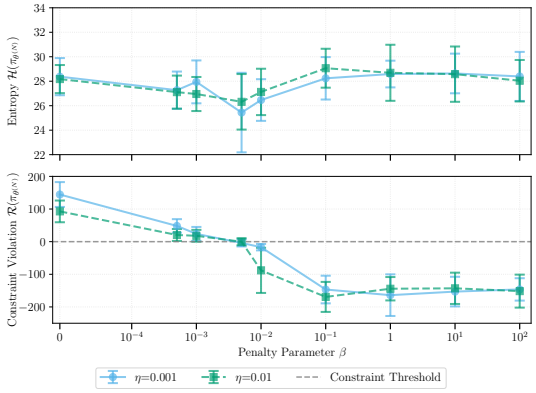


Figure 2: **Robustness to β.** The entropy of the final policy $\pi_{\theta^{(N)}}$ is insensitive to the choice of β across several orders of magnitude, while constraint violations decrease monotonically as β increases, for both **step sizes** $\eta = 0.001$ and $\eta = 0.01$. Experimental details are available in Appendix E.

β	$5 \cdot 10^{-3}$	10^{-2}	10^{-1}	1	10	10^2
Conv. epoch	2200	2800	3500	5000	7000	10000

(a) **Grid world.** Number of epochs until convergence as a function of β, cf. Figure 2.

β	Conv. epoch	Entropy	Constraint
10	2500	798.22 ± 214.13	-12.78 ± 8.42
50	3500	825.82 ± 211.10	-13.94 ± 8.67
100	10000	787.29 ± 246.31	-12.50 ± 7.82
200	14000	817.61 ± 235.52	-13.72 ± 8.12

(b) **SafeCartpole.** Number of epochs until convergence, terminal entropy and constraint values ($\mu \pm \sigma$) over β.

Table 1: **Effect of overestimating β.** Approximate convergence epochs estimated from saved checkpoints. Once β is large enough to enforce feasibility, further increases mainly slow convergence while leaving the final entropy and constraint values essentially unchanged.

Imitation Learning Constraint. We train a reference policy π_{ref} using the Soft-Actor Critic (SAC) method [105] to solve the *PointMass* environment [42]. We then apply our method to (CME) with an apprenticeship constraint $\mathcal{R}(\lambda^{\pi_\theta}) := \mathsf{D}_{\mathsf{KL}}(\lambda^{\pi_\theta} \| \lambda^{\pi_{\text{ref}}}) - \mathcal{R}_{\max} \leq 0$. Figure 3 illustrates the central tension of this setting: the reference policy consistently directs the agent toward the center of the environment, inherently discouraging exploration. As the constraint budget $\mathcal{R}_{\max}$ increases, the learned policy is permitted greater deviation from π_{ref} and progressively overcomes this bias, achieving increasingly uniform coverage of the state space. We visualize the controls for different starting positions and varying $\mathcal{R}_{\max}$ in Figure 14 (Appendix).

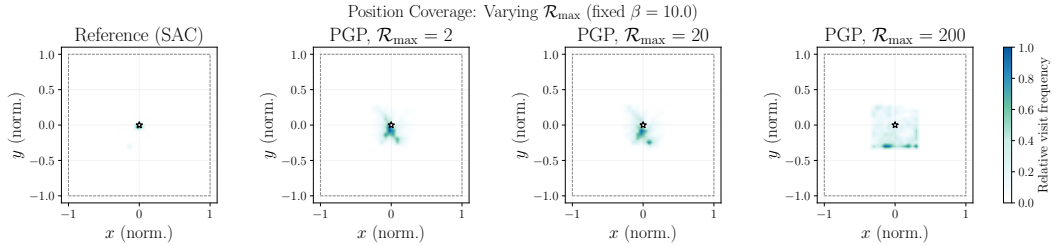
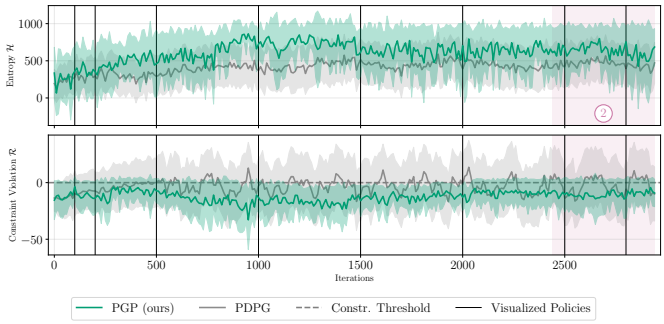


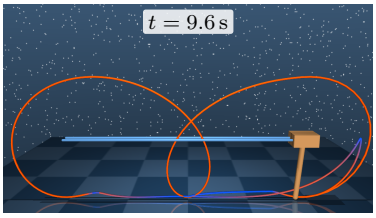
Figure 3: **PointMass: State-Space Coverage.** Relative visit frequencies of the SAC reference policy (trained to reach $(\star)$) and PGP-policies evaluated on 30 start positions. Reference policy concentrates near the center, while increasing $\mathcal{R}_{\max}$ yields progressively more uniform state-space coverage.

Reward Constraint. The *SafeCartpole* task [42] imposes a (linear) symmetric box constraint $\mathcal{R}(\lambda^{\pi_\theta}) := \langle c_{\text{pos}}, \lambda^{\pi_\theta} \rangle - c_{\max} \leq 0$ on the cart position, making exploration structurally challenging. Effective exploration requires a swing-up, which must be executed within tight positional bounds.

Figure 4a quantifies this tension: increasing entropy, necessary for swing-up, systematically drives constraint violations toward the safety threshold, with PGP successfully navigating this tradeoff (Figure 4b). Compared with the tuned PDPG baseline, PGP maintains sustained swing-up and feasibility substantially more consistently over the final 500 epochs, cf. pink region ② in Figure 4a and quantified in Table 2b. This comparison is practical evidence beyond the scope of our finite-space theorem. The policy snapshots (vertical lines in Figure 4a) trace the training dynamics in Figure 17 (Appendix): early iterations focus on swing-up, intermediate policies exploit symmetry for broader state-space coverage, and later stages approach the constraint boundary to minimize swing-up time while maintaining feasibility. A detailed quantitative analysis is provided in Appendix E.3.3.



(a) PGP (ours) and PDPG: entropy $\mathcal{H}$ (top) and constraint violation $\mathcal{R}$ (bottom). Area ② marks the final 500 epochs in Table 2b; vertical lines mark the policies visualized in Figure 17 (Appendix).



(b) Final policy $\pi_{\theta^{(N)}}$ satisfying the cart-position constraint $[-2.0, 2.0]$. Tracer color denotes pole angular velocity (lower, higher), see Appendix E.3.3.

Figure 4: (CME) **on SafeCartpole.** PGP shows safe exploration of the full state space (pos. and vel.), while PDPG suffers under constraint oscillations, cf. Table 2b.

Method	$\mathcal{R} \leq 0.1$	$\mathcal{R} \leq 0.01$
PGP (ours)	**97.21%**	**92.43%**
PDPG	65.33%	51.01%

(a) **Grid world:** final 1k epochs over 10 seeds, (10k checkpts), cf. Fig. 1 ①.

Method	SU $>$ 1 st.	SU $>$ 60 st.	$\mathcal{R} \leq 0.1$	$\mathcal{R} \leq 0.01$
PGP (ours)	**87.23%**	**86.11%**	**96.89%**	**94.49%**
PDPG	59.47%	41.64%	70.03%	53.33%

(b) **SafeCartpole:** final 500 epochs over 5 seeds (2.5k checkpts), cf. Figure 4 ②. "SU" denotes swing-up and "st." denotes steps.

Table 2: **Last-iterate stability.** Fractions of late-training checkpoints satisfying feasibility and swing-up; PGP is more consistently feasible and exploratory than PDPG.

7 Concluding Remarks & Future Work

We propose a single-loop, policy-space penalty method for constrained maximum-entropy exploration. Our analysis provides global last-iterate convergence guarantees under strong duality, despite non-convex policy parameterization and stochastic gradients, and we demonstrate the scalability of our method on a continuous control task. Future work includes extensions to non-smooth occupancy-measure constraints. On the algorithmic side, closing the theory-practice gap in step-size and batch-size selection remains an important and non-trivial challenge. Finally, our current algorithm and analysis yield a relatively large $\widetilde{\mathcal{O}}(\varepsilon^{-6})$ sample complexity. We believe this sample complexity can be reduced by using additional algorithmic refinements, such as variance reduction or adaptive penalty schemes. However, a rigorous validation of these directions remains open.

Acknowledgement. FW is supported by the Prof. Dr.-Ing. Erich Müller-Stiftung, the Almetes Stiftung, the Swiss National Centre of Competence in Research (NCCR) Automation under grant agreement No. 51NF40_180545, and the Kortschak Scholars Program. IF is supported by ETH AI Center Doctoral Fellowship. The authors thank Yarden As for fruitful discussions.

References

[1] J. Achiam, D. Held, A. Tamar, and P. Abbeel. Constrained Policy Optimization, May 2017. arXiv:1705.10528. (Cited on page(s) 2)

[2] M. Agarwal, Q. Bai, and V. Aggarwal. Concave Utility Reinforcement Learning with Zero-Constraint Violations. *Transactions on Machine Learning Research*, Sept. 2022. (Cited on page(s) 2, 3)

[3] S. Agarwal, I. Durugkar, P. Stone, and A. Zhang. f-Policy Gradients: A General Framework for Goal Conditioned RL using f-Divergences, Oct. 2023. arXiv:2310.06794. (Cited on page(s) 21)

[4] E. Altman. *Constrained Markov Decision Processes: Stochastic Modeling*. Routledge, Boca Raton, 1 edition, Dec. 2021. doi: 10.1201/9781315140223. (Cited on page(s) 2)

[5] Y. As, I. Usmanova, S. Curi, and A. Krause. Constrained Policy Optimization via Bayesian World Models. In *International Conference on Learning Representations*, 2022. (Cited on page(s) 2)

[6] Y. As, B. Sukhija, L. Treven, C. Sferrazza, S. Coros, and A. Krause. ActSafe: Active Exploration with Safety Constraints for Reinforcement Learning. In *International Conference on Learning Representations*, 2025. (Cited on page(s) 2)

[7] A. Auslender, J. P. Crouzeix, and P. Fedit. Penalty-proximal methods in convex programming. *Journal of Optimization Theory and Applications*, 55(1):1–21, Oct. 1987. doi: 10.1007/BF00939042. (Cited on page(s) 32)

[8] N. S. Aybat and G. Iyengar. A First-Order Augmented Lagrangian Method for Compressed Sensing. *SIAM Journal on Optimization*, 22(2):429–459, Jan. 2012. doi: 10.1137/100786721. (Cited on page(s) 21, 31)

[9] Q. Bai, A. S. Bedi, M. Agarwal, A. Koppel, and V. Aggarwal. Achieving Zero Constraint Violation for Concave Utility Constrained Reinforcement Learning via Primal-Dual Approach. *Journal of Artificial Intelligence Research*, 78:975–1016, Dec. 2023. doi: 10.1613/jair.1.15383. (Cited on page(s) 2, 3)

[10] Q. Bai, A. S. Bedi, and V. Aggarwal. Achieving Zero Constraint Violation for Constrained Reinforcement Learning via Conservative Natural Policy Gradient Primal-Dual Algorithm. In *AAAI Conference on Artificial Intelligence*, pages 6737–6744, 2023. doi: 10.1609/AAAI.V37I6.25826. arXiv:2206.05850. (Cited on page(s) 2)

[11] A. Barakat, I. Fatkhullin, and N. He. Reinforcement Learning with General Utilities: Simpler Variance Reduction and Large State-Action Space. In *International Conference on Machine Learning*, pages 1753–1800, 2023. (Cited on page(s) 3, 7, 8, 23)

[12] A. Barakat, S. Chakraborty, P. Yu, P. Tokekar, and A. S. Bedi. Towards Scalable General Utility Reinforcement Learning: Occupancy Approximation, Sample Complexity and Global Optimality, Feb. 2025. arXiv:2410.04108. (Cited on page(s) 3, 7, 8, 50, 54)

[13] A. Barakat, A. Kontogiannis, V. Pollatos, I. Panageas, and A. Varvitsiotis. Online Learning on Hidden-Convex Losses via Algorithmic Equivalence: Optimal Regret, Geometric Barrier, and Bandit Feedback, May 2026. arXiv:2605.26373. (Cited on page(s) 3)

[14] D. P. Bertsekas. *Nonlinear Programming*. Athena Scientific, 2 edition, Sept. 1999. (Cited on page(s) 21, 31)

[15] D. P. Bertsekas, editor. *Convex Optimization Theory*. Athena Scientific, Belmont, Mass, 2009. (Cited on page(s) 7)

[16] D. Blessing, O. Celik, X. Jia, M. Reuss, M. X. Li, R. Lioutikov, and G. Neumann. Information maximizing curriculum: A curriculum-based approach for imitating diverse skills. In *Advances in Neural Information Processing Systems*, pages 51536–51561, 2023. (Cited on page(s) 2)

[17] D. Boob, Q. Deng, and G. Lan. Stochastic first-order methods for convex and nonconvex functional constrained optimization. *Mathematical Programming*, 197(1):215–279, Jan. 2023. doi: 10.1007/s10107-021-01742-y. (Cited on page(s) 35)

[18] D. Boob, Q. Deng, and G. Lan. Level constrained first order methods for function constrained optimization. *Mathematical Programming*, 209(1-2):1–61, Jan. 2025. doi: 10.1007/s10107-024-02057-4. (Cited on page(s) 35)

[19] S. Boyd and N. Parikh. Proximal Algorithms. *Found. Trends Optim.*, 1(3):127–239, Jan. 2014. doi: 10.1561/2400000003. (Cited on page(s) 31, 35)

[20] S. P. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, Cambridge, UK ; New York, 2004. (Cited on page(s) 21, 31)

[21] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba. OpenAI Gym, June 2016. arXiv:1606.01540. (Cited on page(s) 58, 60, 62)

[22] M. Buckley, K. Papathanasiou, and A. Spanopoulos. Primal-Dual Sample Complexity Bounds for Constrained Markov Decision Processes with Multiple Constraints, Mar. 2025. arXiv:2503.06751. (Cited on page(s) 2)

[23] X. Chen, Z. Lu, and T. K. Pong. Penalty Methods for a Class of Non-Lipschitz Optimization Problems. *SIAM Journal on Optimization*, July 2016. doi: 10.1137/15M1028054. (Cited on page(s) 31)

[24] Z. Chen, Y. Wen, Z. Hu, and H. Huang. Robust Reinforcement Learning with General Utility. *Advances in Neural Information Processing Systems*, 37:11290–11344, Jan. 2025. (Cited on page(s) 3)

[25] Y. Chow, O. Nachum, E. Duenez-Guzman, and M. Ghavamzadeh. A Lyapunov-based Approach to Safe Reinforcement Learning. In *Advances in Neural Information Processing Systems*, pages 8103–8112, 2018. (Cited on page(s) 2)

[26] R. Courant. Variational methods for the solution of problems of equilibrium and vibrations. *Bulletin of the American Mathematical Society*, 49(1):1–23, Jan. 1943. (Cited on page(s) 21, 31)

[27] Y. Cui, Q. Shi, X. Wang, and X. Xiao. An Exact Penalty Method for Stochastic Equality-Constrained Optimization, Nov. 2025. (Cited on page(s) 6)

[28] G. Dalal, K. Dvijotham, M. Vecerik, T. Hester, C. Paduraru, and Y. Tassa. Safe Exploration in Continuous Action Spaces, Jan. 2018. arXiv:1801.08757. (Cited on page(s) 2)

[29] D. Davis and D. Drusvyatskiy. Stochastic subgradient method converges at the rate $O(K^{-1/4})$ on weakly convex functions, Feb. 2018. arXiv:1802.02988. (Cited on page(s) 35)

[30] D. Davis and D. Drusvyatskiy. Stochastic Model-Based Minimization of Weakly Convex Functions. *SIAM Journal on Optimization*, 29(1):207–239, Jan. 2019. doi: 10.1137/18M1178244. (Cited on page(s) 31, 34)

[31] D. Davis and B. Grimmer. Proximally Guided Stochastic Subgradient Method for Nonsmooth, Nonconvex Problems. *SIAM Journal on Optimization*, 29(3):1908–1930, Jan. 2019. doi: 10.1137/17M1151031. (Cited on page(s) 31, 35)

[32] D. Davis, D. Drusvyatskiy, K. J. MacPhee, and C. Paquette. Subgradient Methods for Sharp Weakly Convex Functions. *Journal of Optimization Theory and Applications*, 179(3):962–982, Dec. 2018. doi: 10.1007/s10957-018-1372-8. (Cited on page(s) 31, 35)

[33] D. Davis, D. Drusvyatskiy, S. Kakade, and J. D. Lee. Stochastic Subgradient Method Converges on Tame Functions. *Foundations of Computational Mathematics*, 20(1):119–154, Feb. 2020. doi: 10.1007/s10208-018-09409-5. arXiv:1804.07795. (Cited on page(s) 31, 35)

[34] D. Davis, D. Drusvyatskiy, and C. Paquette. The nonsmooth landscape of phase retrieval. *IMA Journal of Numerical Analysis*, 40(4):2652–2695, Oct. 2020. doi: 10.1093/imanum/drz031. arXiv:1711.03247. (Cited on page(s) 31)

[35] R. De Santi, M. Prajapat, and A. Krause. Global reinforcement learning: Beyond linear and convex rewards via submodular semi-gradient methods. In *International Conference on Machine Learning*, pages 10235–10266, 2024. (Cited on page(s) 3)

[36] N. K. Dhingra, S. Z. Khong, and M. R. Jovanović. The proximal augmented Lagrangian method for nonsmooth composite optimization. *IEEE Transactions on Automatic Control*, 64 (7):2861–2868, July 2019. doi: 10.1109/TAC.2018.2867589. arXiv:1610.04514. (Cited on page(s) 32)

[37] D. Ding and M. R. Jovanović. Policy gradient primal-dual mirror descent for constrained MDPs with large state spaces. In *2022 IEEE 61st Conference on Decision and Control (CDC)*, pages 4892–4897, 2022. doi: 10.1109/CDC51059.2022.9992419. (Cited on page(s) 2)

[38] D. Ding, K. Zhang, T. Basar, and M. Jovanovic. Natural Policy Gradient Primal-Dual Method for Constrained Markov Decision Processes. In *Advances in Neural Information Processing Systems*, pages 8378–8390, 2020. (Cited on page(s) 2)

[39] D. Ding, C.-Y. Wei, K. Zhang, and A. Ribeiro. Last-Iterate Convergent Policy Gradient Primal-Dual Methods for Constrained MDPs. In *Advances in Neural Information Processing Systems*, 2023. arXiv:2306.11700. (Cited on page(s) 2)

[40] D. Ding, K. Zhang, J. Duan, T. Basar, and M. R. Jovanovic. Convergence and Sample Complexity of Natural Policy Gradient Primal-Dual Methods for Constrained MDPs. *Journal of Machine Learning Research*, 26(256):1–76, 2025. (Cited on page(s) 2)

[41] D. Drusvyatskiy and C. Paquette. Efficiency of minimizing compositions of convex functions and smooth maps. *Mathematical Programming*, 178(1-2):503–558, Nov. 2019. doi: 10.1007/s10107-018-1311-3. (Cited on page(s) 34, 41, 42)

[42] G. Dulac-Arnold, N. Levine, D. J. Mankowitz, J. Li, C. Paduraru, S. Gowal, and T. Hester. Challenges of real-world reinforcement learning: Definitions, benchmarks and analysis. *Machine Learning*, 110(9):2419–2468, Sept. 2021. doi: 10.1007/s10994-021-05961-4. (Cited on page(s) 1, 9, 45, 57, 58, 60, 62)

[43] Y. Efroni, S. Mannor, and M. Pirotta. Exploration-Exploitation in Constrained MDPs, Mar. 2020. arXiv:2003.02189. (Cited on page(s) 2)

[44] I. Fatkhullin, A. Barakat, A. Kireeva, and N. He. Stochastic Policy Gradient Methods: Improved Sample Complexity for Fisher-non-degenerate Policies. In *International Conference on Machine Learning*, pages 9827–9869, 2023. arXiv:2302.01734. (Cited on page(s) 8)

[45] I. Fatkhullin, N. He, and Y. Hu. Stochastic Optimization under Hidden Convexity. *SIAM Journal on Optimization*, pages 2544–2571, Dec. 2025. doi: 10.1137/22M1708903. arXiv:2401.00108. (Cited on page(s) 3, 20, 22, 26, 31, 32, 34, 35)

[46] I. Fatkhullin, N. He, G. Lan, and F. Wolf. Global Solutions to Non-Convex Functional Constrained Problems with Hidden Convexity, Nov. 2025. arXiv:2511.10626. (Cited on page(s) 3, 7, 31, 32, 34, 35, 40, 41)

[47] A. V. Fiacco and G. P. McCormick. *Nonlinear Programming: Sequential Unconstrained Minimization Techniques*. SIAM: Classics in Applied Mathematics. John Wiley & Sons Ltd, New York, NY, 1990. doi: 10.1137/1.9781611971316. (Cited on page(s) 21, 31)

[48] M. Geist, J. Pérolat, M. Laurière, R. Elie, S. Perrin, O. Bachem, R. Munos, and O. Pietquin. Concave Utility Reinforcement Learning: The Mean-field Game Viewpoint. In *International Conference on Autonomous Agents and Multiagent Systems*, pages 489–497, 2022. (Cited on page(s) 3)

[49] A. Ghosh, X. Zhou, and N. B. Shroff. Provably Efficient Model-Free Constrained RL with Linear Function Approximation. In *Advances in Neural Information Processing Systems*, 2022. (Cited on page(s) 2)

[50] Z. D. Guo, M. G. Azar, A. Saade, S. Thakoor, B. Piot, B. A. Pires, M. Valko, T. Mesnard, T. Lattimore, and R. Munos. Geometric Entropic Exploration, Jan. 2021. arXiv:2101.02055. (Cited on page(s) 21)

[51] E. Hazan, S. Kakade, K. Singh, and A. V. Soest. Provably Efficient Maximum Entropy Exploration. In *International Conference on Machine Learning*, pages 2681–2691, 2019. (Cited on page(s) 1, 2, 7)

[52] A. Huang and N. Jiang. Occupancy-based Policy Gradient: Estimation, Convergence, and Optimality. In *Advances in Neural Information Processing Systems*, 2024. (Cited on page(s) 3)

[53] Y. Huang and Q. Lin. Oracle Complexity of Single-Loop Switching Subgradient Methods for Non-Smooth Weakly Convex Functional Constrained Optimization. *Advances in Neural Information Processing Systems*, 36:61327–61340, Dec. 2023. (Cited on page(s) 31, 35)

[54] R. Islam, Z. Ahmed, and D. Precup. Marginalized State Distribution Entropy Regularization in Policy Optimization, Dec. 2019. arXiv:1912.05128. (Cited on page(s) 21)

[55] R. Islamov, Y. As, and I. Fatkhullin. Safe-EF: Error Feedback for Non-smooth Constrained Optimization. In *International Conference on Machine Learning*, 2025. (Cited on page(s) 3)

[56] A. K. Jain, L. Lehnert, I. Rish, and G. Berseth. Maximum State Entropy Exploration using Predecessor and Successor Representations. In *Advances in Neural Information Processing Systems*, 2023. (Cited on page(s) 21)

[57] Z. Jia and B. Grimmer. First-Order Methods for Nonsmooth Nonconvex Functional Constrained Optimization with or without Slater Points. *SIAM J. Optim.*, 35(2):1300–1329, 2025. doi: 10.1137/23M1569551. (Cited on page(s) 31, 35)

[58] P. Jordan, A. Barakat, and N. He. Independent Learning in Constrained Markov Potential Games. In *International Conference on Artificial Intelligence and Statistics*, pages 4024–4032, 2024. (Cited on page(s) 31)

[59] F. Kalogiannis, J. Yan, and I. Panageas. Learning Equilibria in Adversarial Team Markov Games: A Nonconvex-Hidden-Concave Min-Max Optimization Problem. In *Advances in Neural Information Processing Systems*, 2024. arXiv:2410.05673. (Cited on page(s) 3)

[60] D. Kim, J. Shin, P. Abbeel, and Y. Seo. Accelerating Reinforcement Learning with Value-Conditional State Entropy Exploration. *Advances in Neural Information Processing Systems*, 36:31811–31830, Dec. 2023. (Cited on page(s) 2)

[61] P. Klink, F. Wolf, K. Ploeger, J. Peters, and J. Pajarinen. Tracking Control for a Spherical Pendulum via Curriculum Reinforcement Learning, Sept. 2023. arXiv:2309.14096. (Cited on page(s) 2)

[62] W. Kong, J. G. Melo, and R. D. C. Monteiro. Complexity of a Quadratic Penalty Accelerated Inexact Proximal Point Method for Solving Linearly Constrained Nonconvex Composite Programs. *SIAM Journal on Optimization*, 29(4):2566–2593, Jan. 2019. doi: 10.1137/18M1171011. (Cited on page(s) 32)

[63] H. J. Kushner and E. Sanvicente. Penalty function methods for constrained stochastic approximation. *Journal of Mathematical Analysis and Applications*, 46(2):499–512, May 1974. doi: 10.1016/0022-247X(74)90256-X. (Cited on page(s) 6)

[64] G. Lan. First-order and Stochastic Optimization Methods for Machine Learning. In G. Lan, editor, *First-Order and Stochastic Optimization Methods for Machine Learning*, pages 53–111. Springer International Publishing, Cham, 2020. doi: 10.1007/978-3-030-39568-1_3. (Cited on page(s) 38)

[65] G. Lan and R. D. C. Monteiro. Iteration-complexity of first-order penalty methods for convex programming. *Mathematical Programming*, 138(1):115–139, Apr. 2013. doi: 10.1007/s10107-012-0588-x. (Cited on page(s) 21, 31)

[66] G. Lan and R. D. C. Monteiro. Iteration-complexity of first-order augmented Lagrangian methods for convex programming. *Mathematical Programming*, 155(1):511–547, Jan. 2016. doi: 10.1007/s10107-015-0861-x. (Cited on page(s) 31)

[67] G. Lan and Z. Zhou. Algorithms for stochastic optimization with function or expectation constraints. *Computational Optimization and Applications*, 76(2):461–498, June 2020. doi: 10.1007/s10589-020-00179-x. (Cited on page(s) 40)

[68] L. Lee, B. Eysenbach, E. Parisotto, E. Xing, S. Levine, and R. Salakhutdinov. Efficient Exploration via State Marginal Matching, Feb. 2020. arXiv:1906.05274. (Cited on page(s) 21)

[69] G. Li, W. Yu, Y. Yao, W. Tong, Y. Liang, Q. Lin, and T. Yang. A Retention-Centric Framework for Continual Learning with Guaranteed Model Developmental Safety, Apr. 2025. arXiv:2410.03955. (Cited on page(s) 6)

[70] J. Li, A. M.-C. So, and W.-K. Ma. Understanding Notions of Stationarity in Nonsmooth Optimization: A Guided Tour of Various Constructions of Subdifferential for Nonsmooth Functions. *IEEE Signal Processing Magazine*, 37(5):18–31, Sept. 2020. doi: 10.1109/MSP.2020.3003845. (Cited on page(s) 43)

[71] T. Li, Z. Guan, S. Zou, T. Xu, Y. Liang, and G. Lan. Faster algorithm and sharper analysis for constrained Markov decision process. *Operations Research Letters*, 54:107107, May 2024. doi: 10.1016/j.orl.2024.107107. (Cited on page(s) 2)

[72] Z. Li and Y. Xu. Augmented Lagrangian–Based First-Order Methods for Convex-Constrained Programs with Weakly Convex Objective. *INFORMS Journal on Optimization*, 3(4):373–397, Oct. 2021. doi: 10.1287/ijoo.2021.0052. (Cited on page(s) 21, 31)

[73] Z. Li, P.-Y. Chen, S. Liu, S. Lu, and Y. Xu. Rate-improved inexact augmented Lagrangian method for constrained nonconvex optimization. In *International Conference on Artificial Intelligence and Statistics*, pages 2170–2178, 2021. (Cited on page(s) 31)

[74] Z. Li, P.-Y. Chen, S. Liu, S. Lu, and Y. Xu. Stochastic inexact augmented Lagrangian method for nonconvex expectation constrained optimization. *Comput. Optim. Appl.*, 87(1):117–147, 2024. doi: 10.1007/S10589-023-00521-Z. (Cited on page(s) 6)

[75] Q. Lin, R. Ma, and Y. Xu. Complexity of an inexact proximal-point penalty method for constrained smooth non-convex optimization. *Computational Optimization and Applications*, 82(1):175–224, May 2022. doi: 10.1007/s10589-022-00358-y. (Cited on page(s) 32, 35)

[76] H. Liu and P. Abbeel. APS: Active Pretraining with Successor Features. In *International Conference on Machine Learning*, pages 6736–6747, 2021. (Cited on page(s) 2)

[77] H. Liu and P. Abbeel. Behavior From the Void: Unsupervised Active Pre-Training. In *Advances in Neural Information Processing Systems*, pages 18459–18473, 2021. (Cited on page(s) 2)

[78] T. Liu, R. Zhou, D. Kalathil, P. R. Kumar, and C. Tian. Policy Optimization for Constrained MDPs with Provable Fast Global Convergence, Feb. 2022. arXiv:2111.00552. (Cited on page(s) 2)

[79] W. Liu and Y. Xu. A single-loop SPIDER-type stochastic subgradient method for expectation-constrained nonconvex nonsmooth optimization, Sept. 2025. arXiv:2501.19214. (Cited on page(s) 6)

[80] Y.-F. Liu, X. Liu, and S. Ma. On the Nonergodic Convergence Rate of an Inexact Augmented Lagrangian Framework for Composite Convex Programming. *Mathematics of Operations Research*, 44(2):632–650, May 2019. doi: 10.1287/moor.2018.0939. (Cited on page(s) 31)

[81] M. Lutter. Inductive Biases in Machine Learning for Robotics and Control. 2021. doi: 10.26083/tuprints-00020048. (Cited on page(s) 2)

[82] R. Ma, Q. Lin, and T. Yang. Quadratically Regularized Subgradient Methods for Weakly Convex Optimization with Weakly Convex Constraints. In *International Conference on Machine Learning*, pages 6554–6564, 2020. (Cited on page(s) 31, 35)

[83] B. Marin Moreno, M. Brégère, P. Gaillard, and N. Oudjane. MetaCURL: Non-stationary Concave Utility Reinforcement Learning. *Advances in Neural Information Processing Systems*, 37:123091–123126, Dec. 2024. doi: 10.52202/079017-3912. (Cited on page(s) 3)

[84] B. Martinet. Brève communication. Régularisation d'inéquations variationnelles par approximations successives. *Revue française d'informatique et de recherche opérationnelle. Série rouge*, 4(R3):154–158, 1970. (Cited on page(s) 31)

[85] S. Messaoud, B. Mokeddem, Z. Xue, L. Pang, B. An, H. Chen, and S. Chawla. S2AC: Energy-Based Reinforcement Learning with Stein Soft Actor Critic. In *International Conference on Learning Representations*, 2023. (Cited on page(s) 21)

[86] S. Messaoud, S. Charni, E. Bouazza, A. P. Fatideh, and H. Bensmail. Particles Don't Care About Z: Towards Scaling Entropy Estimation of Unnormalized Densities. In *International Conference on Machine Learning*, 2026. (Cited on page(s) 21)

[87] W. U. Mondal and V. Aggarwal. Sample-Efficient Constrained Reinforcement Learning with General Parameterization. In *Advances in Neural Information Processing Systems*, 2024. arXiv:2405.10624. (Cited on page(s) 2)

[88] J. J. Moreau. Proximité et dualité dans un espace hilbertien. *Bulletin de la Société Mathématique de France*, 93:273–299, 1965. (Cited on page(s) 31)

[89] B. M. Moreno, M. Brégère, P. Gaillard, and N. Oudjane. Efficient Model-Based Concave Utility Reinforcement Learning through Greedy Mirror Descent. In *International Conference on Artificial Intelligence and Statistics*, pages 2206–2214, 2024. (Cited on page(s) 3)

[90] B. M. Moreno, K. Eldowa, P. Gaillard, M. Brégère, and N. Oudjane. Online Episodic Convex Reinforcement Learning, May 2025. arXiv:2505.07303. (Cited on page(s) 3)

[91] A. Müller, P. Alatur, V. Cevher, G. Ramponi, and N. He. Truly No-Regret Learning in Constrained MDPs. In *International Conference on Machine Learning*, 2024. arXiv:2402.15776. (Cited on page(s) 3, 8)

[92] M. Mutti and M. Restelli. An Intrinsically-Motivated Approach for Learning Highly Exploring and Fast Mixing Policies. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34 (04):5232–5239, Apr. 2020. doi: 10.1609/aaai.v34i04.5968. (Cited on page(s) 2)

[93] M. Mutti, L. Pratissoli, and M. Restelli. Task-Agnostic Exploration via Policy Gradient of a Non-Parametric State Entropy Estimate. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(10):9028–9036, May 2021. doi: 10.1609/aaai.v35i10.17091. (Cited on page(s) 1, 2)

[94] M. Mutti, R. D. Santi, P. D. Bartolomeis, and M. Restelli. Challenging Common Assumptions in Convex Reinforcement Learning. In *Advances in Neural Information Processing Systems*, 2022. (Cited on page(s) 3)

[95] M. Mutti, R. D. Santi, and M. Restelli. The Importance of Non-Markovianity in Maximum State Entropy Exploration. In *International Conference on Machine Learning*, pages 16223–16239, 2022. (Cited on page(s) 3)

[96] M. Mutti, R. D. Santi, P. D. Bartolomeis, and M. Restelli. Convex Reinforcement Learning in Finite Trials. *Journal of Machine Learning Research*, 24(250):1–42, 2023. (Cited on page(s) 3)

[97] I. Necoara, A. Patrascu, and F. Glineur. Complexity of first-order inexact Lagrangian and penalty methods for conic convex programming. *Optimization Methods and Software*, 34(2): 305–335, Mar. 2019. doi: 10.1080/10556788.2017.1380642. (Cited on page(s) 21, 31)

[98] Y. Nesterov. *Introductory Lectures on Convex Optimization*, volume 87 of *Applied Optimization*. Springer US, Boston, MA, 2004. doi: 10.1007/978-1-4419-8853-9. (Cited on page(s) 44)

[99] Y. Nesterov. *Lectures on Convex Optimization*, volume 137 of *Springer Optimization and Its Applications*. Springer International Publishing, Cham, Switzerland, 2018. doi: 10.1007/978-3-319-91578-4. (Cited on page(s) 7, 21, 31, 39)

[100] T. Ni and M. Kamgarpour. A Safe Exploration Approach to Constrained Markov Decision Processes. In *International Conference on Artificial Intelligence and Statistics*, pages 3592–3600, 2025. (Cited on page(s) 2)

[101] A. Novy and F. Wolf. Verfahren zur Verbrauchsschätzung von Elektrofahrzeugen und dessen Verwendung zum Ermitteln einer Navigationsroute, Jan. 2023. (Cited on page(s) 50)

[102] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Köpf, E. Z. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems*, pages 8024–8035, 2019. (Cited on page(s) 55, 62)

[103] A. Patrascu, I. Necoara, and Q. Tran-Dinh. Adaptive inexact fast augmented Lagrangian methods for constrained convex optimization. *Optim. Lett.*, 11(3):609–626, 2017. doi: 10.1007/S11590-016-1024-6. (Cited on page(s) 31)

[104] M. Prajapat, M. Mutny, M. Zeilinger, and A. Krause. Submodular Reinforcement Learning. In *International Conference on Learning Representations*, 2023. (Cited on page(s) 3)

[105] A. Raffin, A. Hill, A. Gleave, A. Kanervisto, M. Ernestus, and N. Dormann. Stable-Baselines3: Reliable Reinforcement Learning Implementations. *Journal of Machine Learning Research*, 22(268):1–8, 2021. (Cited on page(s) 9, 54, 62)

[106] J. Ramirez, M. Hashemizadeh, and S. Lacoste-Julien. Position: Adopt Constraints Over Penalties in Deep Learning, July 2025. arXiv:2505.20628. (Cited on page(s) 21)

[107] R. T. Rockafellar. *Convex Analysis*. Princeton Mathematical Series. Princeton University Press, Princeton, N. J., 1970. (Cited on page(s) 31)

[108] R. T. Rockafellar. Monotone Operators and the Proximal Point Algorithm. *SIAM Journal on Control and Optimization*, 14(5):877–898, 1976. doi: 10.1137/0314056. (Cited on page(s) 31)

[109] R. T. Rockafellar and R. J. B. Wets. *Variational Analysis*, volume 317 of *Grundlehren Der Mathematischen Wissenschaften*. Springer, Berlin, Heidelberg, 1998. doi: 10.1007/978-3-642-02431-3. (Cited on page(s) 31, 43)

[110] P. P. Santos, A. Sardinha, and F. S. Melo. The Number of Trials Matters in Infinite-Horizon General-Utility Markov Decision Processes. In *International Conference on Machine Learning*, 2025. (Cited on page(s) 3)

[111] Y. Seo, L. Chen, J. Shin, H. Lee, P. Abbeel, and K. Lee. State Entropy Maximization with Random Encoders for Efficient Exploration. In *International Conference on Machine Learning*, pages 9443–9454, 2021. (Cited on page(s) 2)

[112] A. Stuart and F. Wolf. Expressivity In Multimodal Contrastive Learning, Aug. 2026. arXiv:2608.17203. (Cited on page(s) 50)

[113] R. S. Sutton, D. McAllester, S. Singh, and Y. Mansour. Policy Gradient Methods for Reinforcement Learning with Function Approximation. In *Advances in Neural Information Processing Systems*, 1999. (Cited on page(s) 5)

[114] Y. Tassa, Y. Doron, A. Muldal, T. Erez, Y. Li, D. d. L. Casas, D. Budden, A. Abdolmaleki, J. Merel, A. Lefrancq, T. Lillicrap, and M. Riedmiller. DeepMind Control Suite, Jan. 2018. arXiv:1801.00690. (Cited on page(s) 45, 62)

[115] D. Tiapkin, D. Belomestny, D. Calandriello, E. Moulines, R. Munos, A. Naumov, P. Perrault, Y. Tang, M. Valko, and P. Menard. Fast Rates for Maximum Entropy Exploration. In *International Conference on Machine Learning*, pages 34161–34221, 2023. (Cited on page(s) 1, 2)

[116] G. Tiboni, P. Klink, J. Peters, T. Tommasi, C. D'Eramo, and G. Chalvatzaki. Domain Randomization via Entropy Maximization. In *International Conference on Learning Representations*, 2024. (Cited on page(s) 2)

[117] E. Todorov, T. Erez, and Y. Tassa. MuJoCo: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5026–5033, 2012. doi: 10.1109/IROS.2012.6386109. (Cited on page(s) 45, 62)

[118] Q. Tran-Dinh. Proximal alternating penalty algorithms for nonsmooth constrained convex optimization. *Computational Optimization and Applications*, 72(1):1–43, Jan. 2019. doi: 10.1007/s10589-018-0033-z. (Cited on page(s) 32, 35)

[119] J.-P. Vial. Strong and Weak Convexity of Sets and Functions. *Mathematics of Operations Research*, 8(2):231–259, May 1983. doi: 10.1287/moor.8.2.231. (Cited on page(s) 34, 43)

[120] A. Wachi, W. Hashimoto, X. Shen, and K. Hashimoto. Safe Exploration in Reinforcement Learning: A Generalized Formulation and Algorithms. In *Advances in Neural Information Processing Systems*, 2023. (Cited on page(s) 2)

[121] I.-J. Wang and J. C. Spall. Stochastic optimisation with inequality constraints using simultaneous perturbations and penalty functions. *International Journal of Control*, 81(8):1232–1238, Aug. 2008. doi: 10.1080/00207170701611123. (Cited on page(s) 6)

[122] R. Wang and C. Ding. Robins-Monro Augmented Lagrangian Method for Stochastic Convex Optimization, Sept. 2022. arXiv:2208.14019. (Cited on page(s) 6)

[123] H. Wei, X. Liu, and L. Ying. Triple-Q: A Model-Free Algorithm for Constrained Reinforcement Learning with Sublinear Regret and Zero Constraint Violation. In *International Conference on Artificial Intelligence and Statistics*, pages 3274–3307, 2022. (Cited on page(s) 2)

[124] M. Wendl, Y. As, M. Prajapat, A. Pollak, S. Coros, and A. Krause. Safe Exploration via Policy Priors. In *NeurIPS 2025 Workshop: Second Workshop on Aligning Reinforcement Learning Experimentalists and Theorists*, 2025. (Cited on page(s) 2)

[125] F. Wolf. Stochastic Gradient Descent and its Application for Parametrized Boundary Value Problems under Uncertainties. Master's thesis, University of Konstanz, 2021. (Cited on page(s) 34)

[126] F. Wolf, N. Botteghi, U. Fasel, and A. Manzoni. Interpretable and efficient data-driven discovery and control of distributed systems. *Data-Centric Engineering*, 6:e51, 2025. doi: 10.1017/dce.2025.10027. (Cited on page(s) 2)

[127] F. Wolf, A. Carminati, and A. Guglielmi. Spatio-temporal Clustering of PM2.5 in northern Italy using a Bayesian model. In *Methodological and Applied Statistics and Demography I*, pages 218–223, 2025. doi: 10.1007/978-3-031-64346-0_37. (Cited on page(s) 50)

[128] Z. Wu and R. Xu. Risk-sensitive Markov Decision Process and Learning under General Utility Functions, Dec. 2024. arXiv:2311.13589. (Cited on page(s) 3)

[129] N. Xiao, T. Tang, S. Wang, and K.-C. Toh. An Exact Penalty Approach for Equality Constrained Optimization over a Convex Set, May 2025. arXiv:2505.02495. (Cited on page(s) 31)

[130] Y. Xie and S. J. Wright. Complexity of Proximal Augmented Lagrangian for Nonconvex Optimization with Nonlinear Equality Constraints. *Journal of Scientific Computing*, 86(3):38, Feb. 2021. doi: 10.1007/s10915-021-01409-y. (Cited on page(s) 32, 35)

[131] T. Xu, Y. Liang, and G. Lan. CRPO: A New Approach for Safe Reinforcement Learning with Convergence Guarantee. In *International Conference on Machine Learning*, pages 11480–11491, 2021. arXiv:2011.05869. (Cited on page(s) 2, 3)

[132] Y. Xu. Iteration complexity of inexact augmented Lagrangian methods for constrained convex programming. *Mathematical Programming*, 185(1-2):199–244, 2021. doi: 10.1007/S10107-019-01425-9. (Cited on page(s) 31)

[133] M. Yang, G. Li, Q. Hu, Q. Lin, and T. Yang. Single-loop Algorithms for Stochastic Non-convex Optimization with Weakly-Convex Constraints, Apr. 2025. arXiv:2504.15243. (Cited on page(s) 6)

[134] Q. Yang and M. T. J. Spaan. CEM: Constrained Entropy Maximization for Task-Agnostic Safe Exploration. *Proceedings of the AAAI Conference on Artificial Intelligence*, 37(9): 10798–10806, June 2023. doi: 10.1609/aaai.v37i9.26281. (Cited on page(s) 2)

[135] Q. Yang, T. D. Simão, N. Jansen, S. H. Tindemans, and M. T. J. Spaan. Reinforcement Learning by Guided Safe Exploration. Frontiers in Artificial Intelligence and Applications. Sept. 2023. doi: 10.3233/FAIA230598. arXiv:2307.14316. (Cited on page(s) 2)

[136] T.-Y. Yang, J. Rosca, K. Narasimhan, and P. J. Ramadge. Projection-Based Constrained Policy Optimization, Oct. 2020. arXiv:2010.03152. (Cited on page(s) 2)

[137] X. Yang and X. Huang. Partially Strictly Monotone and Nonlinear Penalty Functions for Constrained Mathematical Programs. *Computational Optimization and Applications*, 25(1): 293–311, Apr. 2003. doi: 10.1023/A:1022929826650. (Cited on page(s) 31)

[138] D. Yarats, R. Fergus, A. Lazaric, and L. Pinto. Reinforcement Learning with Prototypical Representations, July 2021. arXiv:2102.11271. (Cited on page(s) 2)

[139] D. Ying, Y. Ding, and J. Lavaei. A Dual Approach to Constrained Markov Decision Processes with Entropy Regularization. In *International Conference on Artificial Intelligence and Statistics*, pages 1887–1909, 2022. arXiv:2110.08923. (Cited on page(s) 2)

[140] D. Ying, Y. Ding, A. Koppel, and J. Lavaei. Scalable Multi-Agent Reinforcement Learning with General Utilities. In *2023 American Control Conference (ACC)*, pages 3977–3982, 2023. doi: 10.23919/ACC55779.2023.10156072. (Cited on page(s) 3)

[141] D. Ying, M. A. Guo, H. Lee, Y. Ding, J. Lavaei, and Z.-J. M. Shen. Policy-based Primal-Dual Methods for Concave CMDP with Variance Reduction. *Journal of Artificial Intelligence Research*, 83, Aug. 2025. doi: 10.1613/jair.1.18129. (Cited on page(s) 1, 2, 3, 7, 8, 9, 46, 50)

[142] M. Yuan, M.-o. Pun, and D. Wang. Rényi State Entropy for Exploration Acceleration in Reinforcement Learning, Mar. 2022. arXiv:2203.04297. (Cited on page(s) 21)

[143] T. Zahavy, B. O'Donoghue, G. Desjardins, and S. Singh. Reward is enough for convex MDPs. In *Advances in Neural Information Processing Systems*, pages 25746–25759, 2021. arXiv:2106.00661. (Cited on page(s) 3)

[144] R. Zamboni, D. Cirino, M. Restelli, and M. Mutti. How to Explore with Belief: State Entropy Maximization in POMDPs. In *International Conference on Machine Learning*, 2024. (Cited on page(s) 2)

[145] R. Zamboni, D. Cirino, M. Restelli, and M. Mutti. The Limits of Pure Exploration in POMDPs: When the Observation Entropy is Enough. *RLJ*, 2:676–692, 2024. (Cited on page(s) 2)

[146] R. Zamboni, M. Mutti, and M. Restelli. Towards Principled Unsupervised Multi-Agent Reinforcement Learning, Oct. 2025. arXiv:2502.08365. (Cited on page(s) 21)

[147] C. Zhang, Y. Cai, L. Huang, and J. Li. Exploration by Maximizing Renyi Entropy for Reward-Free RL Framework. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(12): 10859–10867, May 2021. doi: 10.1609/aaai.v35i12.17297. (Cited on page(s) 21)

[148] H. Zhang, X. Peng, H. Wei, and X. Liu. Safe and Efficient: A Primal-Dual Method for Offline Convex CMDPs under Partial Data Coverage. *Advances in Neural Information Processing Systems*, 37:34239–34269, Dec. 2024. doi: 10.52202/079017-1079. (Cited on page(s) 3)

[149] J. Zhang and Z.-Q. Luo. A Proximal Alternating Direction Method of Multiplier for Linearly Constrained Nonconvex Minimization. *SIAM Journal on Optimization*, 30(3):2272–2302, Jan. 2020. doi: 10.1137/19M1242276. (Cited on page(s) 32)

[150] J. Zhang, A. Koppel, A. S. Bedi, C. Szepesvári, and M. Wang. Variational Policy Gradient Method for Reinforcement Learning with General Utilities. In *Advances in Neural Information Processing Systems*, pages 4572–4583, 2020. (Cited on page(s) 3, 7, 20, 21, 22, 34)

[151] J. Zhang, J. Kim, B. O'Donoghue, and S. Boyd. Sample Efficient Reinforcement Learning with REINFORCE. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(12): 10887–10895, May 2021. doi: 10.1609/aaai.v35i12.17300. (Cited on page(s) 5)

[152] J. Zhang, C. Ni, Z. Yu, C. Szepesvári, and M. Wang. On the Convergence and Sample Efficiency of Variance-Reduced Policy Gradient Method. In *Advances in Neural Information Processing Systems*, pages 2228–2240, 2021. (Cited on page(s) 3, 7, 8, 23)

[153] Z. Zhu, F. Chen, J. Zhang, and Z. Wen. On the Optimal Lower and Upper Complexity Bounds for a Class of Composite Optimization Problems, Aug. 2023. arXiv:2308.06470. (Cited on page(s) 31)

[154] E. Zisselman, I. Lavie, D. Soudry, and A. Tamar. Explore to Generalize in Zero-Shot RL. In *Advances in Neural Information Processing Systems*, 2023. (Cited on page(s) 21)

[155] N. Zolman, C. Lagemann, U. Fasel, J. N. Kutz, and S. L. Brunton. SINDy-RL for interpretable and efficient model-based reinforcement learning. *Nature Communications*, 16(1):10714, Nov. 2025. doi: 10.1038/s41467-025-65738-4. (Cited on page(s) 2)

Contents

1 Introduction **1**

2 Related Work **2**

3 Preliminaries **3**

4 Problem Formulation **4**

5 Policy Gradient Penalty Method **5**

5.1 Theoretical Analysis 6

5.2 Sample Complexity Analysis 7

6 Numerical Experiments **8**

6.1 Theoretical Setting: Grid World 8

6.2 Beyond Theory: Continuous State-Action Space 8

7 Concluding Remarks & Future Work **10**

A Additional Related Work **21**

A.1 Detailed Comparison to Zhang et al. [150] 21

A.2 Detailed Comparison to Fatkhullin et al. [45] 22

B Useful Technical Lemmas **22**

B.1 Bias & Variance of Gradient Estimator, Smooth Case, (In-)Finite Horizon 22

B.2 Convergence of Biased SGD for (PEN) 25

C Softmax Policy Density and Smoothed Entropy **28**

C.1 Anchored softmax density 28

C.2 Smoothed entropy 29

D Unified Penalty Framework for Deterministic Smooth and Non-Smooth Constrained Optimization under Hidden Convexity **31**

D.1 Key Inequalities for Analysis under Hidden Convexity 32

D.2 Biased Stochastic Gradient Descent under Hidden Convexity 33

D.3 Inexact Proximal Point Penalty Method (IPPPM) 34

D.4 PPPM under Hidden Strong Convexity 40

D.5 Technical Lemmas 41

D.6 Strong Duality: Translation of Convergence Guarantees 44

E Experimental Details & Ablation Studies **45**

E.1 Grid World: Linear Performance Constraint 45

E.2 Grid World: Reference Trajectory Constraint – Norm Constraint 46

E.3 Continuous State-Action Space 50

F Supplementary Material Details **58**

F.1 Additional Material Folder: `/1_videos` 58

F.2 Source Code Folder: `/2_code` 58

A Additional Related Work

We provide additional related work and make precise why our contributions are substantively different from the two most closely related works.

Different Entropy Formulations. Other works deal with Rényi entropy formulation [147, 142], geometric aware Shannon entropy [50], f-Divergence entropy regularization [3], and parametrized Stein Variational Gradient descent for entropy-regularized RL [85, 86]. Lee et al. [68], Islam et al. [54] recast exploration in terms of marginal state distribution learning, while Jain et al. [56] focus on entropy maximization of a single trajectory, Zamboni et al. [146] on a multi-agent RL setting, and Zisselman et al. [154] use ensemble agents to ensure exploration at test-time.

Optimization Literature. Our algorithmic framework is most closely related to penalty and augmented Lagrangian methods (ALM) for constrained optimization. Such methods are classical tools with a long history and broad adoption [26, 47, 14, 20, 99], and have been extensively studied from a first-order complexity perspective in both convex and non-convex settings [8, 65, 97, 72]. However, existing analyses typically focus on deterministic optimization, establish asymptotic convergence or convergence to ε-KKT points, and do not directly apply to entropy-based reinforcement learning, where non-convexity arises from policy parameterization and gradients are estimated stochastically.

Recent work has also highlighted naively incorporating constraints as costs in the objective function may fail to simultaneously ensure feasibility and optimality in deep learning settings [106]. Our results demonstrate that, when combined with problem-specific gradient structure, quadratic penalty methods can yield strong guarantees even in stochastic, non-convex RL problems. A unified view connecting our approach to proximal-point and ALM frameworks under hidden convexity is developed in Appendix D.

A.1 Detailed Comparison to Zhang et al. [150]

Problem. Zhang et al. [150] study *unconstrained* general-utility RL, i.e., maximizing a convex functional $F(\lambda^\pi)$ without constraints; introducing constraints changes the optimization landscape fundamentally. Moreover, Example 2.1 therein is restricted to cumulative discounted reward in a standard constrained MDP via a generic penalty function, leaving both the concrete choice of penalty and the specific objective unspecified.

Algorithm. The core algorithmic novelty is not in running biased SGD per se, but in *how* we construct the gradient estimator for the penalized objective. In generic stochastic penalty and ALM methods, the chain rule through $\beta\,[\mathcal{R}(\lambda(\theta))]_+^2$ canonically requires separate stochastic estimators for the constraint function value $\mathcal{R}(\lambda(\theta))$ and its sub-gradient; their product creates a cross-term bias that does not vanish with batch size. This fundamental bias mismatch has forced the prior stochastic penalty and ALM methods to either (i) restrict to deterministic constraints, (ii) use nested sampling or multiple gradient evaluations per iteration, or (iii) rely on momentum or variance-tracking. Our pseudo-reward construction (Line 5 of Algorithm 1) circumvents this entirely by reinterpreting *both* gradient terms as policy gradients with respect to constructed pseudo-rewards via the Policy Gradient Theorem. This construction is specific to the RL structure and does not appear in, and cannot be derived from, Zhang et al. [150], which operates in the unconstrained setting.

Theory. Zhang et al. [150] explicitly assume *exact* knowledge of the Q-function (cf. the Remark after their Theorem 2); our analysis operates in the realistic stochastic setting, requiring a joint control of truncation bias $\mathcal{O}(\gamma^H)$ and Monte-Carlo variance $\mathcal{O}(B^{-1/2})$ (cf. Lemma B.2). Beyond this stochastic extension, two further nontrivial analytical steps are required:

- **Preservation of hidden convexity under the penalty** (Lemma B.3): Showing that composing $\mathcal{R}(\lambda)$ with the quadratic penalty $[\,\cdot\,]_+^2$ preserves the smoothness and hidden convexity structure, and deriving the β-dependent smoothness constants $(\ell_{P,\lambda}, L_{P,\lambda}, L_{P,\lambda,\infty})$, requires careful analysis of the interaction between the penalty nonlinearity and the occupancy-measure geometry. We are not aware of any prior work establishing such a result.
- **Translation under strong duality** (Corollary 5.6, Corollary D.22): Converting the penalty gap $P_{2,\beta}^{\mathsf{opt}}(\theta^{(N)}) \le \varepsilon$ into simultaneous objective and constraint guarantees requires a quantitative relationship between $\beta(\varepsilon)$ and the dual multiplier η^*. This translation, yielding the specific scaling $\beta = \mathcal{O}(\varepsilon^{-1})$, is new to hidden convex optimization.

A.2 Detailed Comparison to Fatkhullin et al. [45]

Fatkhullin et al. [45] analyze *unconstrained* stochastic optimization under hidden convexity with an *unbiased* gradient oracle. Thus constraints, stochastic biased gradients, and their interaction with hidden convexity are not addressed therein. We borrow the hidden-convexity framework but face **two difficulties** absent in Fatkhullin et al. [45].

- First, our gradient estimator introduces bias through two distinct mechanisms: (i) trajectory truncation at horizon H, and (ii) the occupancy-measure estimation error propagated through the penalty gradient via the pseudo-rewards, which are evaluated at the *estimated* occupancy measure $\widehat{\lambda}$ rather than the true λ_H.
- Second, the penalty introduces a composite structure whose chain rule canonically induces two independently estimated quantities which we resolve by introducing our pseudo-reward construction.

Controlling the chain of penalty smoothness preservation, through occupancy-measure error propagation, to the final bias-variance tradeoff, is the technical core of our analysis beyond a direct adaptation of Fatkhullin et al. [45] and requires carefully balancing the horizon H, batch size B, as well as the the step-size η is a three-way trade-off absent from the unbiased oracle model of Fatkhullin et al. [45].

Furthermore, translating the penalty guarantees to the constrained problem requires the scaling $\beta = \mathcal{O}\big(\varepsilon^{-1}\big)$ from strong duality (Assumption 5.5), a condition entirely absent in the unconstrained framework. To the best of our knowledge, no prior work handles the combination of hidden convex constraints and stochastic biased gradients.

B Useful Technical Lemmas

B.1 Bias & Variance of Gradient Estimator, Smooth Case, (In-)Finite Horizon

In this section, we will derive the bias and the variance of our gradient estimator (∇_θ-Est) for $F \in \{\mathcal{H}, \mathcal{R}\}$. For a fixed F in this subsection, we abbreviate its function-specific constants by $\ell_\lambda := \ell_{F,\lambda}$,

$L_\lambda \coloneqq L_{F,\lambda}$, and $L_{\lambda,\infty} \coloneqq L_{F,\lambda,\infty}$, respectively. Appendix D.5 shows that the composition of such a function with our quadratic penalty maintains the important property of L-smoothness and hidden convexity.

Lemma B.1 (Properties of (∇_θ-Est))**.** *Under Assumption 5.1 and Assumption 5.3, the following statements hold:*

- $\forall \theta \in \Theta, \forall (s,a) \in \mathcal{S} \times \mathcal{A}$ *we have*

$$\begin{aligned} \|\nabla_\theta \log \pi_\theta(a|s)\| &\leq 2\ell_\psi \\ \left\|\nabla_\theta^2 \log \pi_\theta(a|s)\right\| &\leq 2(L_\psi + \ell_\psi^2) \\ \|\nabla_\theta F(\lambda(\theta))\| &\leq \frac{2\ell_\psi \ell_\lambda}{1-\gamma} =: \ell_{F,\theta} \end{aligned}$$

- $\forall \theta_1, \theta_2 \in \Theta$ *the (truncated) state-action occupancy measure is Lipschitz with respect to the parametrization, i.e.*

$$\begin{aligned} \|\lambda^{\pi_{\theta_1}} - \lambda^{\pi_{\theta_2}}\|_1 &\leq \frac{2\ell_\psi}{1-\gamma}\|\theta_1 - \theta_2\| \\ \|\lambda_H(\theta_1) - \lambda_H(\theta_2)\|_1 &\leq \frac{2\ell_\psi}{1-\gamma}\|\theta_1 - \theta_2\| \end{aligned}$$

- *Concatenating F with the parametrization maintains smoothness, i.e. the function $\theta \mapsto F(\lambda^{\pi_\theta})$ is $L_{F,\theta}$-smooth with*

$$L_{F,\theta} \coloneqq \frac{4L_{\lambda,\infty}\ell_\psi^2}{(1-\gamma)^2} + \frac{4(1+\gamma)\ell_\lambda \ell_\psi^2}{(1-\gamma)^2} + \frac{2\ell_\lambda(L_\psi + \ell_\psi^2)}{1-\gamma}.$$

Proof. See Lem. H.1 in Barakat et al. [11] and Lem. 5.3 in Zhang et al. [152]. In particular, using

$$(1-\gamma)\sum_{t=0}^{\infty}(t+1)\gamma^t = \frac{1}{1-\gamma}, \quad (1-\gamma)\sum_{t=0}^{\infty}(t+1)^2\gamma^t = \frac{1+\gamma}{(1-\gamma)^2},$$

gives the stated gradient and occupancy Lipschitz bounds. Applying the chain rule to $F \circ \lambda$, the Lipschitz variation of $\nabla_\lambda F$ gives the first term in $L_{F,\theta}$, while the squared-score and score-Hessian terms give the second and third terms, respectively. □

The following lemma allows us to run (biased) SGD as a solver for the unconstrained penalty problem.

Lemma B.2 (Gradient Estimator: Bias and Variance)**.** *Under Assumption 5.1 and Assumption 5.3, the full-batch gradient estimator $\hat{g}_{H,B}(\nabla_\lambda F, \theta, (\tau_i)_{i\in[B]})$ in (∇_θ-Est) satisfies, for every $\theta \in \Theta$,*

$$\begin{aligned} \left\|\mathbb{E}\left[\hat{g}_{H,B}(\nabla_\lambda F, \theta, (\tau_i)_{i\in[B]})\right] - \nabla_\theta F(\lambda(\theta))\right\| &\leq D_H \gamma^H + \frac{D_{\hat{g}}}{\sqrt{B}}, \\ \mathbb{E}\left[\left\|\mathbb{E}\left[\hat{g}_{H,B}(\nabla_\lambda F, \theta, (\tau_i)_{i\in[B]})\right] - \hat{g}_{H,B}(\nabla_\lambda F, \theta, (\tau_i)_{i\in[B]})\right\|^2\right] &\leq \frac{\sigma_H^2}{B}, \end{aligned}$$

where

$$\begin{aligned} D_H &\coloneqq 2\ell_\psi\left[\frac{L_{\lambda,\infty}}{1-\gamma} + \ell_\lambda\left(H+1+\frac{\gamma}{1-\gamma}\right)\right], \\ \kappa_H &\coloneqq 2\ell_\psi(1-\gamma)\sum_{t=0}^{H-1}(t+1)\gamma^t \leq \frac{2\ell_\psi}{1-\gamma}, \\ D_{\hat{g}} &\coloneqq \kappa_H L_\lambda, \\ \sigma_H^2 &\coloneqq 2\kappa_H^2\left(\ell_\lambda^2 + L_\lambda^2\right). \end{aligned}$$

Importantly, the same trajectory batch *is used in the occupancy estimate and in the REINFORCE term, i.e. no independence between these two quantities is assumed.*

Proof. For any fixed pseudo-reward r with $\|r\|_\infty \le \ell_\lambda$, the normalized policy-gradient expansion satisfies

$$\|\mathbb{E}[g_H(\theta,\tau,r)]\| \le \frac{2\ell_\psi}{1-\gamma}\|r\|_\infty.$$

Set $q_H := \nabla_\lambda F(\lambda_H(\theta))$ and $q := \nabla_\lambda F(\lambda(\theta))$. Adding and subtracting $[\nabla_\theta \lambda_H(\theta)]^\top q$ gives

$$\nabla_\theta F(\lambda_H(\theta)) - \nabla_\theta F(\lambda(\theta)) = [\nabla_\theta \lambda_H(\theta)]^\top (q_H - q) + \left([\nabla_\theta \lambda_H(\theta)]^\top - [\nabla_\theta \lambda(\theta)]^\top\right) q.$$

The first term is the pseudo-reward approximation error, while the second is the truncation tail for the fixed pseudo-reward q. Using $\|\lambda - \lambda_H\|_1 = \gamma^H$ for the pseudo-reward error and bounding the omitted fixed-reward tail gives

$$\|\nabla_\theta F(\lambda_H(\theta)) - \nabla_\theta F(\lambda(\theta))\| \le \frac{2\ell_\psi L_{\lambda,\infty}}{1-\gamma}\gamma^H + 2\ell_\psi \ell_\lambda (1-\gamma)\sum_{t=H}^{\infty}(t+1)\gamma^t = D_H \gamma^H.$$

For each trajectory τ_i, define its normalized empirical occupancy and its trajectory-wise linear policy-gradient operator by

$$X_i := (1-\gamma)\sum_{t=0}^{H-1}\gamma^t \delta_{s_t^{(i)}, a_t^{(i)}}, \quad \mathsf{M}_i r := g_H(\theta, \tau_i, r).$$

Let

$$\bar{X}_B := \frac{1}{B}\sum_{i=1}^{B} X_i, \qquad \bar{\mathsf{M}}_B := \frac{1}{B}\sum_{i=1}^{B}\mathsf{M}_i, \qquad \mathsf{M} := \mathbb{E}[\mathsf{M}_i],$$

and with $\bar{X}_B = \widehat{\lambda}((\tau_i)_{i\in[B]})$, $\mathbb{E}[X_i] = \lambda_H$, and for every fixed r, it holds that

$$\mathsf{M} r = [\nabla_\theta \lambda_H(\theta)]^\top r.$$

Moreover, reordering the two sums in g_H and using $\|\nabla_\theta \log \pi_\theta(a \mid s)\| \le 2\ell_\psi$ yields the pathwise operator bound

$$\|\mathsf{M}_i r\| \le 2\ell_\psi(1-\gamma)\sum_{t=0}^{H-1}(t+1)\gamma^t\|r\|_\infty = \kappa_H \|r\|_\infty.$$

Since $\|X_i\|_1 = 1-\gamma^H \le 1$, independence across trajectories gives

$$\begin{aligned}\mathbb{E}\left[\|\bar{X}_B - \lambda_H\|_2^2\right] &= \frac{1}{B}\mathbb{E}\left[\|X_1 - \lambda_H\|_2^2\right] = \frac{1}{B}\left(\mathbb{E}\left[\|X_1\|_2^2\right] - \|\lambda_H\|_2^2\right) \le \frac{1}{B}\mathbb{E}\left[\|X_1\|_1^2\right] \\ &= \frac{(1-\gamma^H)^2}{B} \le \frac{1}{B}. \end{aligned} \tag{1}$$

Bounding the Bias. Set $q(\lambda) := \nabla_\lambda F(\lambda)$. The common pseudo-reward in (∇_θ-Est) gives the exact factorization

$$\hat{g}_{H,B}(\nabla_\lambda F, \theta, (\tau_i)_{i\in[B]}) = \bar{\mathsf{M}}_B q(\bar{X}_B).$$

Consequently,

$$\begin{aligned}\mathbb{E}\left[\bar{\mathsf{M}}_B q(\bar{X}_B)\right] - \nabla_\theta F(\lambda_H(\theta)) &= \mathbb{E}\left[(\bar{\mathsf{M}}_B - \mathsf{M})q(\lambda_H)\right] + \mathbb{E}\left[\bar{\mathsf{M}}_B(q(\bar{X}_B) - q(\lambda_H))\right] \\ &= \mathbb{E}\left[\bar{\mathsf{M}}_B(q(\bar{X}_B) - q(\lambda_H))\right],\end{aligned}$$

because $\mathbb{E}\left[\bar{\mathsf{M}}_B\right] = \mathsf{M}$. Thus, without requiring independence between $\bar{\mathsf{M}}_B$ and $\bar{X}_B$, we obtain the upper bound

$$\|\mathbb{E}[\hat{g}_{H,B}] - \nabla_\theta F(\lambda_H(\theta))\| \le \kappa_H L_\lambda \mathbb{E}\left[\|\bar{X}_B - \lambda_H\|_2\right] \le \frac{\kappa_H L_\lambda}{\sqrt{B}} = \frac{D_{\hat{g}}}{\sqrt{B}}.$$

Combining this estimate with the truncation bound proves the bias claim.

Bounding the Variance. The expected squared distance from the mean is no larger than that from the deterministic vector $\mathsf{M}q(\lambda_H) = \nabla_\theta F(\lambda_H(\theta))$. Therefore,

$$\mathbb{E}\Big[\|\hat{g}_{H,B} - \mathbb{E}[\hat{g}_{H,B}]\|^2\Big] \le \mathbb{E}\Big[\big\|\bar{\mathsf{M}}_B q(\bar{X}_B) - \mathsf{M}q(\lambda_H)\big\|^2\Big] \\ \le 2\mathbb{E}\Big[\big\|(\bar{\mathsf{M}}_B - \mathsf{M})q(\lambda_H)\big\|^2\Big] + 2\mathbb{E}\Big[\big\|\bar{\mathsf{M}}_B(q(\bar{X}_B) - q(\lambda_H))\big\|^2\Big],$$

where the first term is an average of independent, centered, fixed-pseudo-reward policy-gradient operators, so

$$\mathbb{E}\Big[\big\|(\bar{\mathsf{M}}_B - \mathsf{M})q(\lambda_H)\big\|^2\Big] = \frac{1}{B}\mathbb{E}\Big[\|(\mathsf{M}_1 - \mathsf{M})q(\lambda_H)\|^2\Big] \le \frac{\kappa_H^2 \ell_\lambda^2}{B},$$

and the second term can be controlled pathwise using (1), yielding

$$\mathbb{E}\Big[\big\|\bar{\mathsf{M}}_B(q(\bar{X}_B) - q(\lambda_H))\big\|^2\Big] \le \kappa_H^2 L_\lambda^2 \mathbb{E}\big[\|\bar{X}_B - \lambda_H\|_2^2\big] \le \frac{\kappa_H^2 L_\lambda^2}{B}.$$

In total, combining the last three bounds, yields

$$\mathbb{E}\Big[\|\hat{g}_{H,B} - \mathbb{E}[\hat{g}_{H,B}]\|^2\Big] \le \frac{2\kappa_H^2(\ell_\lambda^2 + L_\lambda^2)}{B} = \frac{\sigma_H^2}{B},$$

which proves the variance claim and concludes the proof. □

B.2 Convergence of Biased SGD for (PEN)

We start with the following auxilary result, yielding the Lipschitz constant of the penalty function's gradient.

Lemma B.3 (Smoothness and Bounded Gradients of Penalty function in λ)**.** *With $\mathcal{L} = [\,\cdot\,]_+^2$, under Assumption 5.1 and Assumption 5.3, the function $\lambda \mapsto P(\lambda) = -\mathcal{H}(\lambda) + \beta[\mathcal{R}(\lambda)]_+^2$ fulfills the inequalities of Assumption 5.3 with*

$$\ell_{P,\lambda} \coloneqq \ell_{\mathcal{H},\lambda} + 2\beta\mathcal{R}_{\mathsf{UB}}\ell_{\mathcal{R},\lambda},$$
$$L_{P,\lambda} \coloneqq L_{\mathcal{H},\lambda} + 2\beta\left(\mathcal{R}_{\mathsf{UB}}L_{\mathcal{R},\lambda} + \sqrt{|\mathcal{S}|\,|\mathcal{A}|}\,\ell_{\mathcal{R},\lambda}^2\right),$$
$$L_{P,\lambda,\infty} \coloneqq L_{\mathcal{H},\lambda,\infty} + 2\beta\left(\mathcal{R}_{\mathsf{UB}}L_{\mathcal{R},\lambda,\infty} + \ell_{\mathcal{R},\lambda}^2\right),$$

with the upper bound $\mathcal{R}_{\mathsf{UB}} \coloneqq \max_{\lambda\in\Lambda}|\mathcal{R}(\lambda)|$. In particular, the smoothness constants of the entropy function enter additively and are not multiplied by β.

Proof. Writing $r(\lambda) = [\mathcal{R}(\lambda)]_+$, the chain rule gives

$$\nabla_\lambda P(\lambda) = -\nabla_\lambda\mathcal{H}(\lambda) + 2\beta r(\lambda)\nabla_\lambda\mathcal{R}(\lambda).$$

The gradient bound follows from $0 \le r \le \mathcal{R}_{\mathsf{UB}}$. The penalty-gradient difference is bounded by

$$2\beta r(\lambda)\|\nabla\mathcal{R}(\lambda) - \nabla\mathcal{R}(\lambda')\|_\infty + 2\beta|r(\lambda) - r(\lambda')|\|\nabla\mathcal{R}(\lambda')\|_\infty.$$

The positive-part map is non-expansive, and hence $|r(\lambda) - r(\lambda')| \le \ell_{\mathcal{R},\lambda}\|\lambda - \lambda'\|_1$. For the Euclidean bound, the assumed infinity-norm gradient bound gives $\|\nabla\mathcal{R}\|_2 \le \sqrt{|\mathcal{S}|\,|\mathcal{A}|}\,\ell_{\mathcal{R},\lambda}$. Combining these inequalities with the function-specific smoothness constants proves all three claims. □

Corollary B.4 (Gradient Bound and Smoothness of Penalty Function)**.** *Let Assumption 5.1 and Assumption 5.3 hold. For the penalty function (PEN), we get the following constants for the gradients with respect to $\theta \in \Theta$ in the parameter space:*

$$\ell_{P,\theta} = \frac{2\ell_\psi\ell_{P,\lambda}}{1-\gamma},$$
$$L_{P,\theta} = \frac{4L_{P,\lambda,\infty}\ell_\psi^2}{(1-\gamma)^2} + \frac{4(1+\gamma)\ell_{P,\lambda}\ell_\psi^2}{(1-\gamma)^2} + \frac{2\ell_{P,\lambda}(L_\psi + \ell_\psi^2)}{1-\gamma},$$

as the upper bound of the norm of the gradients and the Lipschitz constant of the gradient, respectively.

Proof. Apply Lemma B.1 to the penalty objective with ℓ_λ, $L_{\lambda,\infty}$ replaced by $\ell_{P,\lambda}$, $L_{P,\lambda,\infty}$ from Lemma B.3. □

Corollary B.5 (Bias & Variance of Penalty Function Gradient Estimator)**.** *Under Assumption 5.1 and Assumption 5.3, the full-batch gradient estimator $\hat{g}_{H,B}(\nabla_\lambda P, \theta, (\tau_i)_{i\in[B]})$ in (∇_θ-Est) satisfies, for every $\theta \in \Theta$,*

$$\left\|\mathbb{E}\left[\hat{g}_{H,B}(\nabla_\lambda P, \theta, (\tau_i)_{i\in[B]})\right] - \nabla_\theta P(\lambda(\theta))\right\| \leq D_{P,H}\gamma^H + \frac{D_{P,\hat{g}}}{\sqrt{B}},$$

$$\mathbb{E}\left[\left\|\mathbb{E}\left[\hat{g}_{H,B}(\nabla_\lambda P, \theta, (\tau_i)_{i\in[B]})\right] - \hat{g}_{H,B}(\nabla_\lambda P, \theta, (\tau_i)_{i\in[B]})\right\|^2\right] \leq \frac{\sigma_{P,H}^2}{B},$$

where

$$\begin{aligned} D_{P,H} &\coloneqq 2\ell_\psi\left[\frac{L_{P,\lambda,\infty}}{1-\gamma} + \ell_{P,\lambda}\left(H+1+\frac{\gamma}{1-\gamma}\right)\right], \\ D_{P,\hat{g}} &\coloneqq \kappa_H L_{P,\lambda}, \\ \sigma_{P,H}^2 &\coloneqq 2\kappa_H^2\left(\ell_{P,\lambda}^2 + L_{P,\lambda}^2\right), \end{aligned}$$

and κ_H is defined in Lemma B.2.

Proof. Apply Lemma B.2 to the penalty objective with the bounds $\ell_{P,\lambda}$, $L_{P,\lambda}$, and $L_{P,\lambda,\infty}$ from Lemma B.3. □

Next, we are ready to prove the main convergence result, namely applying biased SGD on our penalty problem (PEN).

B.2.1 Proof of Theorem 5.4

Proof. The key modification of the proof of Thm. 7 in [45] is to construct an element of the form $\theta_\alpha \coloneqq \lambda_{\mathcal{V}(\theta)}^{-1}\left((1-\alpha)\lambda(\theta) + \alpha\lambda(\theta^*)\right)$, $\theta \in \Theta$, with θ^* being the global solution to our constrained problem (CME). The constant α is a minimum of the value in Theorem D.3 and ε_λ of Assumption 5.2. Invoking Theorem D.3 yields $\mathbb{E}\left[P_{2,\beta}^{\mathsf{opt}}(\theta^{(N+1)})\right] \leq \varepsilon$ after

$$N \geq 3\mathcal{D}_\Lambda^2\mu_\lambda^2\frac{L_{P,\theta}}{\varepsilon}\log\left(\frac{3P_{2,\beta}^{\mathsf{opt}}(\theta^{(0)})}{\varepsilon}\right) = \widetilde{\mathcal{O}}\left(\frac{L_{P,\theta}}{\varepsilon}\right).$$

Our Theorem D.3 requires both, the squared gradient bias and its variance to be at most $\varepsilon^2/(9\mathcal{D}_\Lambda^2\mu_\lambda^2)$ which we achieve by setting

$$B \geq \max\left\{\frac{36\mathcal{D}_\Lambda^2\mu_\lambda^2 D_{P,\hat{g}}^2}{\varepsilon^2}, \frac{9\mathcal{D}_\Lambda^2\mu_\lambda^2\sigma_{P,H}^2}{\varepsilon^2}\right\} = \mathcal{O}\left(\varepsilon^{-2}\right),$$

and

$$D_{P,H}\gamma^H \leq \frac{\varepsilon}{6\mathcal{D}_\Lambda\mu_\lambda},$$

In the notation of Corollary B.5 and by absorbing the fixed hidden-convex geometry constants, we obtain

$$B = \mathcal{O}\left(\frac{L_{P,\lambda}^2 + \ell_{P,\lambda}^2}{(1-\gamma)^2\varepsilon^2}\right).$$

Finally, the exact expression for $D_{P,H}$ is affine in H and the truncation condition is achieved with

$$H = \widetilde{\mathcal{O}}\left(\frac{1}{-\log\gamma}\log\left(\frac{\mathcal{D}_\Lambda\mu_\lambda(L_{P,\lambda,\infty} + \ell_{P,\lambda})}{\varepsilon}\right)\right),$$

which concludes the proof. □

B.2.2 Proof of Corollary 5.6

Proof. The proof is similar to the proof of Theorem D.9. Define $\beta := \beta(\varepsilon) := (\nu^* + 1)\frac{\nu^* + \sqrt{\nu^{*2}+2}}{\varepsilon} = \mathcal{O}(\varepsilon^{-1})$. With a target accuracy of $\tilde{\varepsilon} := \frac{1}{\beta(\varepsilon)} \leq \varepsilon$, Theorem 5.4 guarantees that after $N = \mathcal{O}\left(\frac{1}{\tilde{\varepsilon}} + \frac{\beta}{\tilde{\varepsilon}}\right) = \mathcal{O}(\frac{1}{\varepsilon^2})$ iterations

$$\mathbb{E}\left[P^{\mathsf{opt}}_{2,\beta}(\theta^{(N)})\right] \leq \tilde{\varepsilon} \leq \varepsilon$$

holds. Under strong duality, Assumption 5.5, with an optimal Lagrangian multiplier $\nu^* \geq 0$, Corollary D.22 shows that this implies

$$\mathbb{E}\left[[F_2(\theta^{(N)})]_+\right] \leq \frac{\nu^* + \sqrt{\nu^{*2} + 2\beta(\varepsilon)\tilde{\varepsilon}}}{\beta(\varepsilon)} = \frac{\nu^* + \sqrt{\nu^{*2}+2}}{\beta(\varepsilon)} = \frac{\varepsilon}{\nu^* + 1} \leq \varepsilon.$$

Bounding the corresponding optimality gap for F_1 follows by the fact that

$$\mathbb{E}\left[-F_1(\theta^{(N)}) + F^*\right] \leq \mathbb{E}\left[P^{\mathsf{opt}}_{2,\beta}(\theta^{(N)})\right] \leq \varepsilon$$

as well as

$$-\mathbb{E}\left[-F_1(\theta^{(N)}) + F^*\right] \leq \nu^* \mathbb{E}\left[[F_2(\theta^{(N)})]_+\right] \leq \nu^* \frac{\varepsilon}{\nu^* + 1} \leq \varepsilon,$$

by Lemma D.20. In total we obtain the two-sided bound on the expected optimality gap:

$$-\varepsilon \leq \mathbb{E}\left[F_1(\theta^{(N)}) - F^*\right] \leq \varepsilon.$$

Substituting the batch size and trajectory length required by Theorem 5.4 gives a total sample complexity of $\tilde{\mathcal{O}}(\varepsilon^{-6})$. □

C Softmax Policy Density and Smoothed Entropy

This section contains two independent results. First, we provide a self-contained policy-class example satisfying the parametrization assumptions of Theorem 5.4. Second, we analyze the smoothed entropy $\mathcal{H}_\sigma$ and its sample complexity. The smoothing analysis does not rely on the softmax density result, and vice versa.

C.1 Anchored softmax density

We work with the anchored direct-softmax parametrization in a finite tabular MDP and compute its occupancy-map constants explicitly.

Proof. (of Proposition 4.1) Without loss of generality it suffices to consider $\varepsilon \in (0, 1]$. Fix the same reference action $a_0 \in \mathcal{A}$ for every state and, for an arbitrary stationary policy $\pi^* \in \Pi$, define the full-support approximation

$$\pi^*_\varepsilon(a|s) := \left(1 - \frac{\varepsilon}{2}\right) \pi^*(a|s) + \frac{\varepsilon}{2\,|\mathcal{A}|}.$$

For every $s \in \mathcal{S}$, this construction satisfies $\|\pi^*_\varepsilon(\,\cdot\,|s) - \pi^*(\,\cdot\,|s)\|_1 \le \varepsilon$ and $\pi^*_\varepsilon(a|s) \ge \varepsilon/(2\,|\mathcal{A}|)$ for all $a \in \mathcal{A}$. Set the reference logit to zero and, for $a \neq a_0$, and choose

$$\theta_{s,a} := \log \frac{\pi^*_\varepsilon(a|s)}{\pi^*_\varepsilon(a_0|s)}.$$

Substitution into the anchored softmax shows that it represents π^*_ε exactly. Importantly, this anchoring trick does not remove the reference action: its probability is $\pi_\theta(a_0|s) = [1 + \sum_{a \neq a_0} \exp(\theta_{s,a})]^{-1} > 0$; only its redundant logit coordinate is fixed. We use the rescaling to ensure the mapping between the paramter space and the occupancy measure fulfills Assumption 5.2, cf. Lemma C.1. Moreover, $|\theta_{s,a}| \le \log(2\,|\mathcal{A}|\,/\varepsilon)$, so one may take $R = \log(2\,|\mathcal{A}|\,/\varepsilon) = \mathcal{O}(\log(|\mathcal{A}|\,/\varepsilon))$, concluding the proof. □

Lemma C.1 (Softmax Parametrization fulfills Assumption 5.1 and 5.2)**.** *Let $\min_{s\in\mathcal{S}} \mu_0(s) > 0$, fix a reference action $a_0 \in \mathcal{A}$, and let $\Theta_R = [-R, R]^{\mathcal{S}\times(\mathcal{A}\setminus\{a_0\})}$ for $R > 0$. Define*

$$\psi(s, a; \theta) := \begin{cases} \theta_{s,a}, & a \neq a_0, \\ 0, & a = a_0, \end{cases} \qquad \textit{and} \qquad \pi_\theta(a|s) := \frac{\exp(\psi(s, a; \theta))}{1 + \sum_{a' \neq a_0} \exp(\theta_{s,a'})}.$$

Then the parameter-to-occupancy mapping $\theta \mapsto \lambda^{\pi_\theta}$ satisfies Assumption 5.1 and Assumption 5.2.

Proof. The anchored score parametrization is linear in θ, with $\|\nabla_\theta \psi(s, a; \theta)\| \le 1$ and $\nabla^2_\theta \psi(s, a; \theta) = 0$. Hence, Assumption 5.1 holds with $\ell_\psi = 1$ and $L_\psi = 0$. Let $\Lambda_R := \{\lambda^{\pi_\theta} : \theta \in \Theta_R\}$. Since the discounted state occupancy satisfies

$$d^{\pi_\theta}(s) := \sum_{a\in\mathcal{A}} \lambda^{\pi_\theta}(s, a) \ge (1 - \gamma)\mu_0(s) > 0,$$

the occupancy measure uniquely determines the policy and its anchored logits through

$$\pi_\theta(a|s) = \frac{\lambda^{\pi_\theta}(s, a)}{\sum_{a'} \lambda^{\pi_\theta}(s, a')}, \qquad \theta_{s,a} = \log \frac{\lambda^{\pi_\theta}(s, a)}{\lambda^{\pi_\theta}(s, a_0)} \quad (a \neq a_0).$$

This proves injectivity. To obtain a uniform Lipschitz inverse, note that every component of every $\lambda \in \Lambda_R$ is bounded below by

$$m_R := (1 - \gamma) \min_{s\in\mathcal{S}} \mu_0(s) \frac{\exp(-R)}{1 + (|\mathcal{A}| - 1)\exp(R)} > 0.$$

The logarithm is $(1/m_R)$-Lipschitz on $[m_R, 1]$; therefore the displayed inverse is Lipschitz on Λ_R with a constant depending only on R, γ, μ_0, $|\mathcal{S}|$, and $|\mathcal{A}|$. Finally, Λ_R is exactly the intersection occupancy measure polytope with the linear inequalities

$$\exp(-R)\lambda(s, a_0) \le \lambda(s, a) \le \exp(R)\lambda(s, a_0), \qquad s \in \mathcal{S},\ a \neq a_0.$$

Consequently, Λ_R is convex. Taking $\mathcal{U}_\theta = \Theta_R$ and $\mathcal{V}_\theta = \Lambda_R$ gives a bijection with a Lipschitz inverse, and every convex combination of λ^{π_θ} with an optimal occupancy in Λ_R remains in Λ_R. This verifies all parts of Assumption 5.2, concluding the proof. □

Theorem C.2 (Final Complexity Guarantee under Anchored Direct Softmax Parametrization)**.** *Let λ^* be the occupancy measure of an optimal stationary, potentially deterministic policy $\pi^* \in \Pi$ of (CME). Let $\varepsilon > 0$ be the target accuracy. Under Assumption 5.3 for the constraint and strong duality, and using the anchored direct-softmax parametrization on $\Theta_R = [-R, R]^{\mathcal{S}\times(\mathcal{A}\setminus\{a_0\})}$, run Algorithm 1 with a step size $\eta = \mathcal{O}(\varepsilon)$, a batch size $B = \mathcal{O}(\varepsilon^{-4})$, a penalty parameter $\beta = \mathcal{O}(\varepsilon^{-1})$, and a trajectory length $H = \mathcal{O}(\log(1/\varepsilon))$. Then*

$$-\varepsilon \le \mathbb{E}[\mathcal{H}(\lambda^{\pi_{\theta^{(N)}}}) - \mathcal{H}(\lambda^*)] \le \varepsilon, \qquad \mathbb{E}[[\mathcal{R}(\lambda^{\pi_{\theta^{(N)}}})]_+] \le \varepsilon$$

after $N = \widetilde{\mathcal{O}}(\varepsilon^{-2})$ iterations, resulting in a total of $\widetilde{\mathcal{O}}(\varepsilon^{-6})$ trajectory rollouts of length $\mathcal{O}(\log(1/\varepsilon))$.

Proof. For any two stationary policies, their normalized discounted occupancy measures satisfy

$$\|\lambda^{\pi} - \lambda^{\pi'}\|_1 \le \frac{1}{1-\gamma} \max_{s\in\mathcal{S}} \|\pi(\cdot\,|s) - \pi'(\cdot\,|s)\|_1.$$

Indeed, the changes in the discounted state occupancy and in the conditional policy are bounded by $\frac{\gamma\rho}{1-\gamma}$ and ρ, respectively, whenever the maximum policy distance on the right-hand side is at most ρ.

(1) Density argument. Choose an approximation accuracy $\varepsilon_{\text{app}} = \widetilde{\mathcal{O}}(\varepsilon)$ sufficiently small. By Proposition 4.1, there are $R = \mathcal{O}(\log(|\mathcal{A}|/\varepsilon_{\text{app}}))$ and $\bar{\pi} \in \{\pi_\theta : \theta \in \Theta_R\}$ whose maximum statewise ℓ_1-distance to π^* is at most ε_{app}. Hence, writing $\bar{\lambda} = \lambda^{\bar{\pi}}$ and $u = \varepsilon_{\text{app}}/(1-\gamma)$,

$$\|\bar{\lambda} - \lambda^*\|_1 \le u.$$

By Lemma C.1, this anchored policy class satisfies Assumption 5.1 and Assumption 5.2.

(2) Running PGP. The entropy objective is uniformly continuous on the finite-dimensional occupancy simplex. In particular, for $d := |\mathcal{S}|\,|\mathcal{A}|$, we have

$$|\mathcal{H}(\bar{\lambda}) - \mathcal{H}(\lambda^*)| = \mathcal{O}(u\log(d/u)).$$

Moreover, Lipschitz continuity of the constraint and $\mathcal{R}(\lambda^*) \le 0$ give $[\mathcal{R}(\bar{\lambda})]_+ \le \ell_{\mathcal{R},\lambda} u$. Therefore, for the penalty objective $P = -\mathcal{H} + \beta[\mathcal{R}]_+^2$ we have

$$P(\bar{\lambda}) - P(\lambda^*) \le \mathcal{O}(u\log(d/u)) + \beta\ell_{\mathcal{R},\lambda}^2 u^2 = \mathcal{O}(\varepsilon),$$

where the last equality follows by taking $\varepsilon_{\text{app}} = \widetilde{\mathcal{O}}(\varepsilon)$ and $\beta = \mathcal{O}(\varepsilon^{-1})$. Thus the policy-approximation and optimization errors can be split at the target scale. Since the minimum of P over Θ_R is no larger than $P(\bar{\lambda})$, the penalty-gap argument of Theorem 5.4, followed by the strong-duality translation in Corollary 5.6, gives the claimed objective and constraint bounds.

(3) Complexity. Every occupancy component is bounded away from zero on the compact class Θ_R, so the entropy constants required by Assumption 5.3 are finite. Using the constants and the convention of Corollary 5.6, the algorithm requires $\eta = \mathcal{O}(\varepsilon)$, $B = \mathcal{O}(\varepsilon^{-4})$, $\beta = \mathcal{O}(\varepsilon^{-1})$, $H = \mathcal{O}(\log(1/\varepsilon))$, and $N = \widetilde{\mathcal{O}}(\varepsilon^{-2})$. Multiplying N and B yields a total of $\widetilde{\mathcal{O}}(\varepsilon^{-6})$ trajectory rollouts. □

C.2 Smoothed entropy

Lemma C.3 (Smoothed entropy, penalty constants, and dual translation)**.** *Let $d := |\mathcal{S}|\,|\mathcal{A}|$ and $\sigma \in (0,1]$. For every $\lambda \in \mathbb{R}_+^d$ with $\|\lambda\|_1 \le 1$, the surrogate the smoothed surrogate $\mathcal{H}_\sigma(\lambda) := -\sum_{s,a} \lambda(s,a)\log\big(\lambda(s,a)+\sigma\big)$ is concave and satisfies*

$$\nabla_i(-\mathcal{H}_\sigma)(\lambda) = \log(\lambda_i + \sigma) + \frac{\lambda_i}{\lambda_i + \sigma}, \tag{2}$$

$$\nabla_{ii}^2(-\mathcal{H}_\sigma)(\lambda) = \frac{\lambda_i + 2\sigma}{(\lambda_i+\sigma)^2}. \tag{3}$$

Consequently, $-\mathcal{H}_\sigma$ satisfies the three regularity inequalities used in Assumption 5.3 with

$$\ell_{\mathcal{H}_\sigma,\lambda} := 1 + \log\frac{2}{\sigma}, \qquad L_{\mathcal{H}_\sigma,\lambda} = L_{\mathcal{H}_\sigma,\lambda,\infty} := \frac{2}{\sigma}. \tag{4}$$

In particular, (2) equals $\log \sigma$ *at an empirical zero. Moreover, with*

$$a_\sigma := \log(1 + d\sigma) \le d\sigma, \tag{5}$$

we have the uniform approximation bound

$$0 \le \mathcal{H}(\lambda) - \mathcal{H}_\sigma(\lambda) \le a_\sigma.$$

For $P_\sigma = -\mathcal{H}_\sigma + \beta[\mathcal{R}]_+^2$*, its constants can therefore be chosen as*

$$\ell_{P_\sigma,\lambda} \le 1 + \log \frac{2}{\sigma} + 2\beta \mathcal{R}_{\mathsf{UB}} \ell_{\mathcal{R},\lambda}, \tag{6}$$

$$L_{P_\sigma,\lambda} \le \frac{2}{\sigma} + 2\beta \left(\mathcal{R}_{\mathsf{UB}} L_{\mathcal{R},\lambda} + \sqrt{d}\, \ell_{\mathcal{R},\lambda}^2 \right), \tag{7}$$

$$L_{P_\sigma,\lambda,\infty} \le \frac{2}{\sigma} + 2\beta \left(\mathcal{R}_{\mathsf{UB}} L_{\mathcal{R},\lambda,\infty} + \ell_{\mathcal{R},\lambda}^2 \right). \tag{8}$$

Finally, let λ^* *be an optimizer of the original (unsmoothed) problem (CME), let* ν^* *be the multiplier from Assumption 5.5, and set* $r(\lambda) = [\mathcal{R}(\lambda)]_+$*. Then the same original strong-duality assumption implies, that for every occupancy* $\lambda = \lambda^{\pi_\theta}$ *generated by some* $\theta \in \Theta$*, it holds*

$$-\mathcal{H}_\sigma(\lambda) + \mathcal{H}_\sigma(\lambda^*) + \beta r(\lambda)^2 \ge \beta r(\lambda)^2 - \nu^* r(\lambda) - a_\sigma. \tag{9}$$

Thus, if λ *satisfies* $\mathbb{E}\big[-\mathcal{H}_\sigma(\lambda) + \mathcal{H}_\sigma(\lambda^*) + \beta r(\lambda)^2\big] \le \delta$ *and* $s := \mathbb{E}[r(\lambda)]$*, then*

$$s \le \frac{\nu^* + \sqrt{\nu^{*2} + 4\beta(\delta + a_\sigma)}}{2\beta}, \tag{10}$$

$$\mathcal{H}(\lambda^*) - \mathbb{E}[\mathcal{H}(\lambda)] \le \delta + a_\sigma, \quad \textit{and} \quad \mathbb{E}[\mathcal{H}(\lambda)] - \mathcal{H}(\lambda^*) \le \nu^* s. \tag{11}$$

Proof. Differentiating each coordinate of $-\mathcal{H}_\sigma(\lambda) = \sum_i \lambda_i \log(\lambda_i + \sigma)$ gives (2)–(3). The Hessian is diagonal and positive semidefinite, and

$$0 \le \frac{x + 2\sigma}{(x + \sigma)^2} \le \frac{2}{\sigma} \qquad (x \ge 0).$$

The mean-value theorem then gives both smoothness inequalities. Since $x + \sigma \in [\sigma, 2]$ for $x \in [0, 1]$, the gradient formula gives $\|\nabla \mathcal{H}_\sigma\|_\infty \le 1 + \log(2/\sigma)$.

For the approximation bound, define $f_\sigma(x) = x \log(1 + \sigma/x)$ for $x > 0$ and $f_\sigma(0) = 0$. Direct differentiation gives $f_\sigma''(x) = -\sigma^2/[x(x + \sigma)^2] \le 0$. For $g(m) = m \log(1 + d\sigma/m)$, $g'(m) = \log(1 + t) - t/(1 + t) \ge 0$ with $t = d\sigma/m$. Thus, if $m = \|\lambda\|_1 > 0$, concavity of f_σ and $m \le 1$ give

$$\mathcal{H}(\lambda) - \mathcal{H}_\sigma(\lambda) = \sum_i f_\sigma(\lambda_i) \le m \log\left(1 + \frac{d\sigma}{m}\right) \le \log(1 + d\sigma) \le d\sigma.$$

The case $m = 0$ follows by continuity. Substitution of (4) into Lemma B.3 proves (6)- (8).

Strong duality for the original problem gives, for every occupancy $\lambda = \lambda^{\pi_\theta}$ with $\theta \in \Theta$, the inequality

$$-\mathcal{H}(\lambda) + \nu^* \mathcal{R}(\lambda) \ge d(\nu^*) = -\mathcal{H}(\lambda^*).$$

Since $\mathcal{R}(\lambda) \le r(\lambda)$, this implies the pointwise bound $\mathcal{H}(\lambda) - \mathcal{H}(\lambda^*) \le \nu^* r(\lambda)$. Writing $q_\sigma = \mathcal{H} - \mathcal{H}_\sigma \in [0, a_\sigma]$ therefore yields

$$\mathcal{H}_\sigma(\lambda) - \mathcal{H}_\sigma(\lambda^*) \le \nu^* r(\lambda) + a_\sigma,$$

which is (9). Taking expectations and applying Jensen's inequality yields $\beta s^2 - \nu^* s \le \delta + a_\sigma$; solving this quadratic proves (10). Finally, using $q_\sigma \in [0, a_\sigma]$ and the nonnegativity of the penalty gives the first inequality in (11); the second is the original dual bound, concluding the proof. □

Corollary C.4 (Sample complexity with $\mathcal{H}_\sigma$)**.** *Let* $\varepsilon \in (0, 1]$*,* $d := |\mathcal{S}|\,|\mathcal{A}|$*, and set*

$$\sigma := \frac{\varepsilon}{4d}, \qquad \delta := \frac{\varepsilon}{4}, \qquad \beta := (\nu^* + 1)\frac{\nu^* + \sqrt{\nu^{*2} + 2}}{\varepsilon}.$$

Under Assumptions 5.1, 5.2, 5.3, and the original strong-duality Assumption 5.5, run Algorithm 1 with the surrogate $\mathcal{H}_\sigma$ and target penalty accuracy δ. Then

$$-\varepsilon \le \mathbb{E}[\mathcal{H}(\lambda^{\pi_{\theta^{(N)}}}) - \mathcal{H}(\lambda^*)] \le \varepsilon, \qquad \mathbb{E}[[\mathcal{R}(\lambda^{\pi_{\theta^{(N)}}})]_+] \le \varepsilon.$$

For fixed discount factor, constraint-regularity constants, and policy-parametrization constants, this requires $N = \widetilde{\mathcal{O}}\big((d+1)\varepsilon^{-2}\big)$ iterations, $B = \widetilde{\mathcal{O}}\big((d+1)^2\varepsilon^{-4}\big)$ trajectories per iteration, and $H = \mathcal{O}(\log(d/\varepsilon))$. Hence, for fixed d, the total number of trajectory rollouts remains $\widetilde{\mathcal{O}}\big(\varepsilon^{-6}\big)$. Tracking dimension explicitly gives the conservative dependence $\widetilde{\mathcal{O}}\big((d+1)^3\varepsilon^{-6}\big)$.

Proof. The choice of σ gives $a_\sigma \le d\sigma = \varepsilon/4$, and Theorem 5.4 gives the smoothed penalty gap with $\delta = \varepsilon/4$. Let $q = \varepsilon/(\nu^* + 1)$. Since $\sqrt{\nu^{*2} + 2} \ge (\nu^* + 1)/2$ for every $\nu^* \ge 0$, the stated choice of β satisfies

$$\beta q^2 - \nu^* q = \varepsilon \frac{\sqrt{\nu^{*2} + 2}}{\nu^* + 1} \ge \frac{\varepsilon}{2} \ge \delta + a_\sigma.$$

Therefore (10) gives $s \le q \le \varepsilon$, while (11) gives the claimed two-sided entropy bound. For the convergence rate, Lemma C.3 gives

$$\ell_{\mathcal{H}_\sigma,\lambda} \le 1 + \log \frac{8d}{\varepsilon}, \qquad L_{\mathcal{H}_\sigma,\lambda,\infty} = L_{\mathcal{H}_\sigma,\lambda} = \frac{8d}{\varepsilon}.$$

Since $\beta = \mathcal{O}\big(\varepsilon^{-1}\big)$, the separated penalty constants satisfy $\ell_{P_\sigma,\lambda} = \widetilde{\mathcal{O}}\big(\varepsilon^{-1}\big)$ and $L_{P_\sigma,\lambda}, L_{P_\sigma,\lambda,\infty} = \mathcal{O}\big((d+1)\varepsilon^{-1}\big)$ up to fixed constraint-regularity constants. Consequently, $L_{P_\sigma,\theta} = \mathcal{O}\big((d+1)\varepsilon^{-1}\big)$, $D_{P,\hat{g}} = \mathcal{O}\big((d+1)\varepsilon^{-1}\big)$, and $\sigma^2_{P,H} = \widetilde{\mathcal{O}}\big((d+1)^2\varepsilon^{-2}\big)$. Substitution into Theorem 5.4 yields the stated values of N, B, and H, and multiplying N and B proves the rollout bound. □

D Unified Penalty Framework for Deterministic Smooth and Non-Smooth Constrained Optimization under Hidden Convexity

This section places the proposed Policy Gradient Penalty (PGP) method into a broader optimization perspective. We show that the constrained maximum-entropy exploration problem studied in the main paper is a special instance of a wider class of constrained optimization problems exhibiting *hidden convexity*, and that our analysis extends naturally to both smooth and non-smooth settings.

In particular, we connect our approach to penalty and augmented Lagrangian methods (ALM), proximal-point methods, and recent developments in (non-)convex optimization under hidden convexity. While these connections are not required to understand or apply the RL algorithm proposed in the main paper, they clarify the generality of our framework and highlight how problem-specific structure enables guarantees that are unavailable to generic penalty or ALM schemes.

Proximal-point Methods and Penalty Methods. Proximal-point methods (PPM), originally introduced by Moreau [88] and Martinet [84], were later systematically developed and popularized by Rockafellar [107, 108], Rockafellar and Wets [109], with Boyd and Parikh [19] providing a recent overview. In the context of non-smooth and non-convex optimization, PPM appears either implicitly in convergence analyses or explicitly as an algorithmic design principle for unconstrained problems [32, 31, 30, 33, 34, 45], and more recently in primal-only methods for solving non-smooth, non-convex (deterministic) constrained optimization problems [23, 82, 53, 57, 46]. Related ideas have also been explored in multi-agent and game-theoretic settings, e.g., [58] for constrained Markov games.

Penalty and augmented Lagrangian methods are classical tools for constrained optimization, dating back to early variational approaches [26, 47], and have since been widely adopted and studied [14, 137, 20, 99, 129]. A substantial body of work analyzes first-order penalty and ALM schemes with explicit complexity guarantees in convex optimization [8, 65, 103, 66, 97, 80, 132, 153], as well as more recently in non-convex settings [72, 73].

ALM-PPM Hybrids and Hidden Convexity. The combination of augmented Lagrangian schemes with proximal-point ideas, originally introduced by Auslender et al. [7], has proven particularly effective. Recent work establishes strong complexity guarantees for smooth non-convex optimization [62, 149, 130, 75], as well as for non-smooth but convex problems [36, 118].

As already mentioned in the main part of this work, (CME) is a special instance of a broader class of (deterministic) optimization problems, called *hidden convex* problems, of the form

$$\min_{\theta\in\Theta} F_1(\theta) \coloneqq H_1(c(\theta)), \quad \text{s.t. } F_2(\theta) \coloneqq H_2(c(\theta)) \leq 0, \tag{HC}$$

where we use the function $c(\,\cdot\,)$ to indicate a more general transformation, beyond the state-action-occupancy measure we studied in the main part of this work. The following central assumption on $c(\,\cdot\,)$ is a stronger version of Assumption 5.2, but the analysis can also be carried out under Assumption 5.2, cf. Fatkhullin et al. [46] after Definition 1.

Definition D.1 (Hidden Convexity)**.** The above problem (HC) is called *hidden convex* with modulus $\mu_c > 0$, if its components satisfy the following underlying conditions.

1. The domain $\Lambda = c(\Theta)$ is convex, the functions $H_1, H_2 : \Lambda \to \mathbb{R}$ are convex, i.e. satisfy for $i = 1, 2$ and for all $u, v \in \Lambda$ and any $\alpha \in [0, 1]$

$$H_i((1-\alpha)u + \alpha v) \leq (1-\alpha)H_i(u) + \alpha H_i(v) - \frac{(1-\alpha)\alpha\mu_H}{2}\|u - v\|^2. \tag{HC-1}$$

   Additionally, we assume (HC) admits a solution $u^* \in \Lambda$ with its corresponding objective function value $F^* \coloneqq H_1(u^*) = F_1(c^{-1}(u^*))$. Throughout this appendix, F^* denotes the minimum which corresponds to $-F^*$ from (CME).

2. The map $c : \Theta \to \Lambda$ is invertible and there exists a $\mu_c > 0$ such that for all $\theta, \tilde{\theta} \in \Theta$ it holds

$$\|c(\theta) - c(\tilde{\theta})\| \geq \mu_c\|\theta - \tilde{\theta}\|. \tag{HC-2}$$

If $\mu_H > 0$, we refer to (HC) as (μ_c, μ_H)-strongly convex. Note that in this more generic setting, we have $\mu_c = 1/\mu_\lambda$ of Assumption 5.2. We can map (CME) to (HC) via $H_1 = -\mathcal{H}$ and $H_2 = \mathcal{R}$.

Note that the condition (HC-2) along with Assumption D.4 imply that the domain Θ has bounded diameter $\mathcal{D}_\Theta \leq \frac{1}{\mu_c}\mathcal{D}_\Lambda$. For necessary and sufficient conditions for (HC-2), we refer the reader to Fatkhullin et al. [45].

We start this section with two key inequalities for the analysis of gradient-based methods under hidden convexity, then we analyze biased SGD under hidden convexity and afterwards we generalize our penalty method efforts of the main part to non-smooth problems.

D.1 Key Inequalities for Analysis under Hidden Convexity

We state the following proposition which is key for deriving global convergence guarantees under hidden convexity.

Proposition D.2 (Fatkhullin et al. [45] Prop. 3)**.** *Let $F_i(\,\cdot\,)$, $i = 1, 2$, be hidden convex with $\mu_c > 0$. For any $\alpha \in [0, 1]$ and $\theta, \tilde{\theta} \in \Theta$, define $\theta_\alpha \coloneqq c^{-1}((1-\alpha)c(\theta) + \alpha c(\tilde{\theta}))$, then, for $i = 1, 2$, the following functional inequality*

$$F_i(\theta_\alpha) \leq (1-\alpha)F_i(\theta) + \alpha F_i(\tilde{\theta}) - \frac{(1-\alpha)\alpha\mu_H}{2}\|c(\theta) - c(\tilde{\theta})\|^2, \tag{HC-FI}$$

and norm inequality

$$\|\theta_\alpha - \theta\| \leq \frac{\alpha}{\mu_c}\|c(\theta) - c(\tilde{\theta})\|. \tag{HC-NI}$$

hold.

Proof. The proof is originally presented in Fatkhullin et al. [45] but we recall it here for educational purposes, since it illustrates the core idea of convergence analyses under hidden convexity.

By (strong) convexity of $H_i(\,\cdot\,)$ and the convexity of the domain Λ, we have

$$\begin{aligned}
F_i(\theta_\alpha) &= F_i(c^{-1}((1-\alpha)c(\theta) + \alpha c(\tilde{\theta}))) \\
&= H_i((1-\alpha)c(\theta) + \alpha c(\tilde{\theta})) \\
&\leq (1-\alpha)H_i(c(\theta)) + \alpha H_i(c(\tilde{\theta})) - \frac{(1-\alpha)\alpha\mu_H}{2}\|c(\theta) - c(\tilde{\theta})\|^2 \\
&= (1-\alpha)F_i(\theta) + \alpha F_i(\tilde{\theta}) - \frac{(1-\alpha)\alpha\mu_H}{2}\|c(\theta) - c(\tilde{\theta})\|^2,
\end{aligned}$$

using that $(1-\alpha)c(\theta) + \alpha c(\tilde{\theta}) \in \Lambda$ for any $\theta \in \Theta$, with $i = 1, 2$.

Using the property of c we obtain

$$\|\theta_\alpha - \theta\| = \|c^{-1}((1-\alpha)c(\theta) + \alpha c(\tilde{\theta})) - c^{-1}(c(\theta))\| \leq \frac{1}{\mu_c}\|\alpha(c(\theta) - c(\tilde{\theta}))\|,$$

which concludes the proof. □

D.2 Biased Stochastic Gradient Descent under Hidden Convexity

To solve our penalty problem (OptGap), we use (projected) Stochastic Gradient Descent (SGD) with a biased gradient estimate. We analyze biased SGD in our more general, hidden convex setting.

Consider the following general update of projected stochastic gradient descent:

$$\theta^{(k+1)} \coloneqq \operatorname{proj}_\Theta\left(\theta^{(k)} - \eta g^{(k)}\right), \quad g^{(k+1)} \coloneqq \hat{g}^{(k+1)}(\theta^{(k)}, \xi^{(k)}), \tag{SGD}$$

where the stochastic gradient oracle satisfies

$$\mathbb{E}_\xi\left[\hat{g}^{(k)}(\theta, \xi)\right] = \nabla_\theta F(\theta) + b(\theta) \tag{SGD-Bias}$$

$$\mathbb{E}_\xi\left[\left\|\hat{g}^{(k)}(\theta, \xi) - \mathbb{E}_\xi\left[\hat{g}^{(k)}(\theta, \xi)\right]\right\|^2\right] \leq \sigma^2 \tag{SGD-Var}$$

with variance $\sigma^2 \geq 0$ and a uniformly bounded bias $\|b(\theta)\|^2 \leq \delta^2$, $\delta \geq 0$, for all $\theta \in \Theta$.

Theorem D.3 (Biased SGD, Hidden Convexity)**.** *Let F be L-smooth and hidden convex with $F^* \coloneqq F(\theta^*)$ for an $\theta^* \in \Theta$, i.e. $F_2 \equiv 0$. Then running (projected) biased (SGD), with a step size $\eta \leq \frac{1}{L}$, yields for every $K \in \mathbb{N}$ and any $\alpha \in [0, 1]$ the following inequality*

$$\mathbb{E}\left[F(\theta^{(N+1)}) - F^*\right] \leq (1-\alpha)^N \Delta_0 + \frac{\mathcal{D}_\Lambda^2}{\mu_c^2 \eta}\alpha + \frac{\eta}{2}\frac{1}{\alpha}\left(\sigma^2 + \delta^2\right)$$

with $\Delta_0 \coloneqq \mathbb{E}\left[F(\theta^{(0)}) - F^*\right]$. In particular, for an arbitrary target accuracy of $\varepsilon > 0$, without loss of generality $\varepsilon \leq \frac{3\mathcal{D}_\Lambda^2 L}{\mu_c^2}$, by picking

$$\alpha \leq \frac{\mu_c^2 \eta}{\mathcal{D}_\Lambda^2}\frac{\varepsilon}{3} \leq \frac{\mu_c^2}{\mathcal{D}_\Lambda^2 L}\frac{\varepsilon}{3}$$

after

$$N \geq \frac{1}{\alpha}\log\left(\frac{3\Delta_0}{\varepsilon}\right) \geq \frac{3\mathcal{D}_\Lambda^2 \cdot L}{\mu_c^2}\frac{1}{\varepsilon}\log\left(\frac{3\Delta_0}{\varepsilon}\right)$$

steps, the last iterate satisfies

$$\mathbb{E}\left[F(\theta^{(N+1)}) - F^*\right] \leq \frac{2\varepsilon}{3} + \frac{\eta}{2}\frac{1}{\alpha}\left(\sigma^2 + \delta^2\right).$$

Consequently, we need the bias and the variance to satisfy

$$\sigma^2, \delta^2 \leq \frac{\varepsilon}{3}\frac{\alpha}{\eta} \leq \frac{\mu_c^2}{9\mathcal{D}_\Lambda^2}\varepsilon^2$$

in order to get ε-close to the global minima.

Setting	Method	Sub-gradient Complexity		Constraint Qualification
		Smooth	Non-smooth	
$F_2(\cdot) \equiv 0$	GD / IPPM+SM	ε^{-1} [150]	ε^{-3} [46]	–
$F_2(\cdot) \not\equiv 0$	IPPPM Algorithm 2 with $\mathcal{L} = [\cdot]_+^2$	ε^{-2} Corollary D.12	ε^{-6} Corollary D.13	Strong Duality
$F_2(\cdot) \not\equiv 0$	IPPPM Algorithm 2 with $\mathcal{L} = [\cdot]_+$	–	ε^{-3} Corollary D.11	Strong Duality

Table 3: Summary of total **(sub-)gradient** and **function evaluation** complexities for deterministic gradient methods under hidden convexity. The second column shows the number of (sub)-gradient (and function in case $F_2(\,\cdot\,) \not\equiv 0$) evaluations in $\widetilde{\mathcal{O}}(\,\cdot\,)$ notation to find a point $\theta^{(N)}$ with $|F_1(\theta^{(N)}) - F^*| \leq \varepsilon$, $[F_2(\theta^{(N)})]_+ \leq \varepsilon$. "IPPPM" refers to the inexact proximal point penalty framework, "GD" to gradient descent established by Zhang et al. [150], and "IPPM+SM" to the inexact proximal point framework combined with the sub-gradient method analyzed by Fatkhullin et al. [46].

Algorithm 2 INEXACT PROXIMAL POINT PENALTY METHOD (IPPPM)

1: **Input:** objective F_1, constraint F_2, target tolerance $\varepsilon > 0$, inner tolerance $\varepsilon_{\text{in}} > 0$, initial point $\theta_0 \in \mathcal{X}$, outer loops N, inner loops T_{in}, inner algorithm $\mathcal{A}$, regularization $\hat{\rho} > \rho_\varphi$, penalty parameter $\beta > 0$, penalty function $\mathcal{L} \in \{[\cdot]_+^2, [\cdot]_+\}$
2: **for** $k = 0$ **to** $N - 1$ **do**
3: Define the regularized penalty function for $\theta \in \Theta$ as

$$\varphi_{\text{pen}}^{(k)}(\theta) := F_1(\theta) + \frac{\hat{\rho}}{2}\|\theta - \theta^{(k)}\|^2 + \beta\mathcal{L}(F_2(\theta)),$$

4: Compute approximate solution of (IPPPM) via

$$\theta^{(k+1)} \leftarrow \mathcal{A}(\varphi_{\text{pen}}^{(k)}, \theta^{(k)}, T_{\text{in}}, \varepsilon_{\text{in}})$$

5: **end for**
6: **Return:** $x^{(N)}$

Proof. We use Thm. 7 in Fatkhullin et al. [45] and that the gradient estimation error can be uniformly bounded by

$$\mathbb{E}\left[\left\|\hat{g}^{(k)} - \nabla F(\theta)\right\|^2\right] = \mathbb{E}\left[\left\|\hat{g}^{(k)} - \mathbb{E}\left[\hat{g}^{(k)}\right]\right\|^2\right] + \mathbb{E}\left[\left\|\mathbb{E}\left[\hat{g}^{(k)}\right] - \nabla F(\theta)\right\|^2\right] \leq \sigma^2 + \delta^2.$$

Plugging this bound into the proof of Thm. 7 in Fatkhullin et al. [45], unrolling the recursion and choosing α as proposed concludes the proof. □

D.3 Inexact Proximal Point Penalty Method (IPPPM)

Inspired by the success of the penalty method for solving (CME), the question arises if penalty methods can be used to solve the broader class of constrained hidden convex optimization problems.

Assumption D.4. We conduct our analysis under the following (standard) assumptions:

- (Weak Convexity) F_1, F_2 to be ρ-weakly convex, for an $\rho > 0$. It is known that weak convexity is a relatively mild assumption, compared to e.g. smoothness [30, 41, 125]. F_i is weakly convex, if e.g. F_i is twice continuously differentiable with the Hessian satisfying $\nabla^2 F_i(\theta) \succcurlyeq -\rho$, for all $\theta \in \Theta$, $i = 1, 2$, cf. Prop 4.12 in Vial [119].
- (Bounded [Sub-]Gradients) The functions F_i, $i = 1, 2$, are continuous and satisfy for all $\theta \in \Theta$ that $\partial F_i(\theta) \neq \emptyset$ and with $\|g\| \leq G$ we have a uniform bound for all sub-gradients $g \in \partial F_i(\theta)$, $\theta \in \Theta$.
- (Bounded Domain) The Domain Λ has a bounded diameter $\mathcal{D}_\Lambda > 0$.

In particular in the context of non-convex, non-smooth optimization, the proximal point method (PPM) framework [19] has proven to be very successful in the unconstrained setting [32, 29–31, 33, 45] as well as in the constrained setting [82, 17, 53, 18, 57, 46], either implicitly in the analysis or explicitly for algorithm design. Motivated by the non-smooth literature, as well as recent advances in proximal point method penalty [75] and augmented Lagrangian methods [118, 130], we propose the following, **inexact proximal point penalty method**:

$$\theta^{(k)} \approx \hat{\theta}^{(k)} \coloneqq \operatorname*{arg\,min}_{\theta\in\Theta} \varphi^{(k)}(\theta) \coloneqq F_1(\theta) + \frac{\hat{\rho}}{2}\|\theta - \theta^{(k)}\|^2 + \beta\mathcal{L}(F_2(\theta)), \tag{IPPPM}$$

where the penalty function $\mathcal{L}$ depends on the smoothness of the constraint. In the case of non-smooth F_1, F_2, the exact penalty function $[\,\cdot\,]_+$ is a natural choice. Under the additional assumption of L-smooth F_1, F_2 one needs to "smoothen" the penalty, in order to propagate the smoothness and thus a quadratic penalty $[\,\cdot\,]_+^2$ is applied. For the convergence analysis of IPPM, we use the notation $\mathcal{L}(\,\cdot\,)$ for both penalty functions and perform a unified analysis by using that both penalty functions are convex and monotone, before we compute the corresponding sample complexities for $[\,\cdot\,]_+$ and $[\,\cdot\,]_+^2$ individually.

Under Assumption D.4, if $\hat{\rho} > \rho$, the inner penalty problem is strongly convex, see Prop. 2.1 in Davis and Grimmer [31].

The **roadmap** of the PPPM analysis is the following:

- *Preservation of Weak Convexity and Bounded (Sub-)Gradients.* Under Assumption D.4, we show in Appendix D.5 that the penalty term $\mathcal{L} \circ F_2$ has uniformly bounded (sub-)gradients and is weakly convex. Thus, (IPPPM) is ρ_φ-weakly convex and can be lifted to be strongly convex. The constant ρ_φ depends on the penalty function and the smoothness assumption.
- *Solve inner problem efficiently.* We apply an efficient algorithm for the inner (non-)smooth strongly convex unconstrained optimization problems to obtain convergence guarantees for the optimality gap of the penalty function $P^{\mathsf{opt}}_{\mathcal{L},\beta}$.
- *Translate guarantees.* Under the assumption of *strong duality*, we translate the results into guarantees on the optimality gap and constraint violation. An overview of all of the resulting rates can be found in Table 3.

Assumption D.5 (Unconstrained Solution Oracle)**.** We assume to have access to an inner approximate solution oracle $\mathcal{A}_{\varepsilon_{\text{in}}}$ to solve (IPPPM) to an ε_{in}-accuracy in expectation for any $\varepsilon_{\text{in}} > 0$, i.e.

$$\varphi^{(k)}_{\text{pen}}(\theta^{(k)}) - \varphi^{(k)}_{\text{pen}}(\hat{\theta}^{(k)}) \leq \varepsilon_{\text{in}}$$

for all iterations $k \in [N]$ after $T_{\text{in}} = T_{\text{in}}(\varepsilon_{\text{in}})$ steps.

Exemplarily, for the non-smooth case, we will formulate Corollary D.11 and Corollary D.13 in terms of the classical (stochastic) sub-gradient descent and in the smooth case Corollary D.12 using SGD, respectively.

D.3.1 Analysis of PPPM, Penalty Function Optimality Value Gap

We start our analysis with the following key lemma, yielding a guarantee on the optimal value gap of the penalty function, not requiring any constraint qualification.

Theorem D.6 (PPPM, Penalty Function)**.** *Let* (HC) *be hidden convex and Assumption D.4 hold. Given a target precision $\varepsilon > 0$, a lifting parameter $\hat{\rho} > \rho_\varphi$, assume $\varepsilon \leq \frac{3\hat{\rho}\mathcal{D}_\Lambda^2}{2\mu_c^2}$ holds. Then after $N \in \mathbb{N}$ steps, the last iterate of Algorithm 2 satisfies*

$$F_1(\theta^{(N)}) - F^* + \beta\mathcal{L}(F_2(\theta^{(N)})) \leq \varepsilon \tag{OptGap}$$

after

$$N \geq \frac{3\hat{\rho}\mathcal{D}_\Lambda^2}{2\mu_c^2}\frac{1}{\varepsilon} \cdot \log\left(\frac{3(\Delta_0 + \beta\mathcal{V}_0)}{\varepsilon}\right)$$

iterations, where $\Delta_0 \coloneqq F_1(\theta^{(0)}) - F^*$, $\mathcal{V}_0 \coloneqq \mathcal{L}(F_2(\theta^{(0)}))$, an $\alpha \leq \frac{2\mu_c^2}{3\hat{\rho}\mathcal{D}_\Lambda^2}\varepsilon \in [0,1]$ used in the analysis and a penalty parameter $\beta > 1$. With an inner precision of $\varepsilon_{\text{in}} \leq \frac{\alpha}{2}\varepsilon$, the total sample

complexity is given by

$$\begin{aligned} T_{\mathrm{tot}} &\geq N \cdot T_{\mathrm{in}}(\varepsilon_{\mathrm{in}}) \\ &= \frac{3\hat{\rho}\mathcal{D}_\Lambda^2}{2\mu_c^2}\frac{1}{\varepsilon} \cdot \log\left(\frac{3(\Delta_0 + \beta\mathcal{V}_0)}{\varepsilon}\right) \cdot T_{\mathrm{in}}(\varepsilon_{\mathrm{in}}). \end{aligned}$$

Proof. We define for an iteration $k \in [N]$ the element $\theta_\alpha^{(k)} \coloneqq c^{-1}((1-\alpha)c(\theta^{(k)}) + \alpha c(\theta^*))$ and we obtain the following inequalities:

$$\begin{aligned} F_1(\theta^{(k+1)}) + \beta\mathcal{L}(F_2(\theta^{(k+1)})) &\overset{(i)}{\leq} \varphi_{\mathrm{pen}}^{(k)}(\theta^{(k+1)}) \overset{(ii)}{\leq} \varphi_{\mathrm{pen}}^{(k)}(\hat{\theta}^{(k+1)}) + \varepsilon_{\mathrm{in}} \overset{(iii)}{\leq} \varphi_{\mathrm{pen}}^{(k)}(\theta_\alpha^{(k)}) + \varepsilon_{\mathrm{in}} \\ &= F_1(\theta_\alpha^{(k)}) + \frac{\hat{\rho}}{2}\|\theta_\alpha^{(k)} - \theta^{(k)}\|^2 + \beta\mathcal{L}(F_2(\theta_\alpha^{(k)})) + \varepsilon_{\mathrm{in}} \\ &\overset{(iv)}{\leq} (1-\alpha)F_1(\theta^{(k)}) + \alpha F^* + \frac{\hat{\rho}}{2}\frac{\alpha^2}{\mu_c^2}\|c(\theta^{(k)}) - c(\theta^*)\|^2 \\ &\qquad + \beta\mathcal{L}((1-\alpha)F_2(\theta^{(k)}) + \alpha F_2(\theta^*)) + \varepsilon_{\mathrm{in}} \\ &\overset{(v)}{\leq} (1-\alpha)F_1(\theta^{(k)}) + \alpha F^* + \alpha^2\frac{\hat{\rho}\mathcal{D}_\Lambda^2}{2\mu_c^2} + \beta(1-\alpha)\mathcal{L}(F_2(\theta^{(k)})) + \varepsilon_{\mathrm{in}} \\ &= (1-\alpha)\Big(F_1(\theta^{(k)}) + \beta\mathcal{L}(F_2(\theta^{(k)}))\Big) + \alpha F^* + \alpha^2\frac{\hat{\rho}\mathcal{D}_\Lambda^2}{2\mu_c^2} + \varepsilon_{\mathrm{in}}, \end{aligned}$$

where we used in (i) that the regularization is always non-zero, in (ii) the $\varepsilon_{\mathrm{in}}$-approximation property of the oracle $\mathcal{A}_{\varepsilon_{\mathrm{in}}}$, in (iii) the optimality of $\hat{\theta}^{(k+1)}$, in (iv) hidden convexity and monotonicity of $\mathcal{L}$, and in (v) that the domain is bounded by our assumption, $F_2(\theta^*) \leq 0$ as well as the fact that $\mathcal{L}(\,\cdot\,)$ is convex.

Subtracting F^* from both sides yields

$$\begin{aligned} &F_1(\theta^{(k+1)}) - F^* + \beta\mathcal{L}(F_2(\theta^{(k+1)})) \\ &\qquad \leq (1-\alpha)\cdot F_1(\theta^{(k)}) - F^* + \beta\mathcal{L}(F_2(\theta^{(k)})) + \alpha^2\frac{\hat{\rho}\mathcal{D}_\Lambda^2}{2\mu_c^2} + \varepsilon_{\mathrm{in}}. \end{aligned} \tag{IPPPM-Rec}$$

We unroll for $k = 0, \ldots, N$ and obtain after $N \in \mathbb{N}$ iterations the following inequalities for the last iterate

$$\begin{aligned} &F_1(\theta^{(N)}) - F^* + \beta\mathcal{L}(F_2(\theta^{(N)})) \\ &\qquad \leq (1-\alpha)^N(\Delta_0 + \beta\mathcal{V}_0) + \alpha\frac{\hat{\rho}\mathcal{D}_\Lambda^2}{2\mu_c^2} + \frac{\varepsilon_{\mathrm{in}}}{\alpha}, \end{aligned}$$

with $\Delta_0 \coloneqq F_1(\theta^{(0)}) - F^*$, and $\mathcal{V}_0 \coloneqq \mathcal{L}(F_2(\theta^{(0)}))$, since the partial sum of the geometric series is bounded by $1/\alpha$. The claim follows by the choice of $\alpha, \varepsilon_{\mathrm{in}}$ and N. □

Remark D.7 (Role of the Penalty Parameter). It is important to emphasize that, although the parameter β appears only logarithmically in the number of outer iterations N, it plays a crucial role in shaping the properties of the penalty function. In particular, β affects the norm of the (sub-)gradients, the associated smoothness constant, and consequently the weak convexity parameter. As a result, it directly influences the computational complexity of the inner solver employed for the subproblem.

Theorem D.8 (PPPM, Exact Penalty)**.** *We assume the assumptions of Theorem D.6 and strong duality hold for* (HC) *with a Lagrangian multiplier $\nu^* \geq 0$. Then, running Algorithm 2 with the exact penalty $\mathcal{L}(\,\cdot\,) = [\,\cdot\,]_+$ and any $\beta \geq 2\cdot\nu^* + 2$ yields following last iterate guarantee on the optimality gap and constraint violation*

$$\left|F_1(\theta^{(N)}) - F^*\right| \leq \varepsilon \quad \textit{and} \quad [F_2(\theta^{(N)})]_+ \leq \varepsilon \tag{IPPPM-Ex}$$

after

$$N \geq \frac{3\hat{\rho}\mathcal{D}_\Lambda^2}{2\mu_c^2}\frac{\beta}{\varepsilon} \cdot \log\left(\frac{3\beta[\Delta_0 + \beta\mathcal{V}_0]}{\varepsilon}\right)$$

iterations, where $\Delta_0 \coloneqq F_1(\theta^{(0)}) - F^*$, *and* $\mathcal{V}_0 \coloneqq [F_2(\theta^{(0)})]_+$. *With an inner precision* $\varepsilon_{\text{in}} \leq \frac{\alpha}{2\beta}\varepsilon$, *where* $\alpha \leq \frac{2\mu_c^2}{3\hat{\rho}\mathcal{D}_\Lambda^2}\varepsilon \in [0,1]$ *as in Theorem D.6, the total rate is given by*

$$\begin{aligned} T_{\text{tot}} &\geq N \cdot T_{\text{in}}(\varepsilon_{\text{in}}) = \frac{3\hat{\rho}\mathcal{D}_\Lambda^2}{2\mu_c^2}\frac{\beta}{\varepsilon} \cdot \log\left(\frac{3\beta[\Delta_0 + \mathcal{V}_0]}{\varepsilon}\right) \cdot T_{\text{in}}(\varepsilon_{\text{in}}) \\ &\geq \frac{3\hat{\rho}\mathcal{D}_\Lambda^2}{2\mu_c^2}\frac{\beta}{\varepsilon} \cdot \log\left(\frac{3\beta[\Delta_0 + \mathcal{V}_0]}{\varepsilon}\right) \cdot T_{\text{in}}\left(\frac{\alpha}{2\beta}\varepsilon\right) \end{aligned}$$

Proof. We define $\tilde{\varepsilon} \coloneqq \frac{\varepsilon}{\beta} < \varepsilon$. By invoking Theorem D.6 with $\tilde{\varepsilon}$, we obtain $P_{1,\beta}^{\mathsf{opt}}(\theta^{(N)}) \leq \tilde{\varepsilon}$ after

$$N \geq \frac{3\hat{\rho}\mathcal{D}_\Lambda^2}{2\mu_c^2}\frac{\beta}{\varepsilon} \cdot \log\left(\frac{3\beta[\Delta_0 + \beta\mathcal{V}_0]}{\varepsilon}\right).$$

Thus, the inequality

$$F_1(\theta^{(N)}) - F^* \leq P_{1,\beta}^{\mathsf{opt}}(\theta^{(N)}) \leq \tilde{\varepsilon} \leq \varepsilon$$

follows trivially and applying Corollary D.21 yields

$$[F_2(\theta^{(N)})]_+ \leq \frac{P_{1,\beta}^{\mathsf{opt}}(\theta^{(N)})}{\beta - \nu^*} \leq \frac{\tilde{\varepsilon}}{\beta - \nu^*} \leq \tilde{\varepsilon} \leq \varepsilon.$$

To obtain the guarantee on the function value gap in absolute value, we derive

$$\begin{aligned} &-\beta[F_2(\theta^{(N)})]_+ \leq F_1(\theta^{(N)}) - F^* \\ \Rightarrow &-\left(F_1(\theta^{(N)}) - F^*\right) \leq \beta[F_2(\theta^{(N)})]_+ \leq \varepsilon \end{aligned}$$

which concludes the proof. □

Theorem D.9 (PPPM, Quadratic Penalty)**.** *We assume the assumptions of Theorem D.6 and strong duality hold for* (HC) *with a Lagrangian multiplier* $\nu^* \geq 0$. *Then, running Algorithm 2 with the quadratic penalty* $\mathcal{L}(\,\cdot\,) = [\,\cdot\,]_+^2$ *and* $\beta = \beta(\varepsilon) \geq \frac{(\nu^*+1)(\nu^*+\sqrt{\nu^{*2}+2})}{\varepsilon}$ *yields following guarantee on the optimality gap and constraint violation*

$$|F_1(\theta^{(N)}) - F^*| \leq \varepsilon \quad \textit{and} \quad [F_2(\theta^{(N)})]_+ \leq \varepsilon \tag{IPPPM-Qu}$$

after

$$N \geq \frac{3\hat{\rho}\mathcal{D}_\Lambda^2}{2\mu_c^2}\beta(\varepsilon) \cdot \log(3\beta(\varepsilon)[\Delta_0 + \beta(\varepsilon)\mathcal{V}_0])$$

iterations, where $\Delta_0 \coloneqq F_1(\theta^{(0)}) - F^*$, *and* $\mathcal{V}_0 \coloneqq [F_2(\theta^{(0)})]_+^2$. *With an inner precision* $\varepsilon_{\text{in}} \leq \frac{\alpha}{2\beta(\varepsilon)}$, *where* $\alpha \leq \frac{2\mu_c^2}{3\hat{\rho}\mathcal{D}_\Lambda^2}\varepsilon \in [0,1]$ *as in Theorem D.6, the total rate is in general given by*

$$\begin{aligned} T_{\text{tot}} &\geq N \cdot T_{\text{in}}(\varepsilon_{\text{in}}) \\ &= \frac{3\hat{\rho}\mathcal{D}_\Lambda^2}{2\mu_c^2}\beta(\varepsilon) \cdot \log(3\beta(\varepsilon)[\Delta_0 + \beta(\varepsilon)\mathcal{V}_0]) \cdot T_{\text{in}}(\varepsilon_{\text{in}}). \end{aligned}$$

Proof. We define $\tilde{\varepsilon} \coloneqq \frac{1}{\beta(\varepsilon)} \leq \varepsilon$. By invoking Theorem D.6 with $\tilde{\varepsilon}$ we obtain $P_{2,\beta}^{\mathsf{opt}}(\theta^{(N)}) \leq \tilde{\varepsilon} \leq \varepsilon$, after

$$N \geq \frac{3\hat{\rho}\mathcal{D}_\Lambda^2}{2\mu_c^2}\beta(\varepsilon) \cdot \log\left(3\beta(\varepsilon)[\Delta_0 + \frac{\beta(\varepsilon)}{2}\mathcal{V}_0]\right)$$

iterations. Trivially, $F_1(\theta^{(N)}) - F^* \leq \varepsilon$ as before. Next, we apply Corollary D.22 to obtain

$$[F_2(\theta^{(N)})]_+ \leq \frac{\nu^* + \sqrt{\nu^{*2} + 2\beta(\varepsilon)\tilde{\varepsilon}}}{\beta(\varepsilon)} = \frac{\nu^* + \sqrt{\nu^{*2}+2}}{\beta(\varepsilon)} = \frac{\varepsilon}{\nu^*+1} \leq \varepsilon,$$

by the choice of $\tilde{\varepsilon}$ and $\beta(\varepsilon)$. To obtain a guarantee on the absolute value of the function optimality value gap, we apply Lemma D.20 to obtain

$$-\left(F_1(\theta^{(N)}) - F^*\right) \leq \nu^*[F_2(\theta^{(N)})]_+ \leq \nu^*\frac{\varepsilon}{\nu^*+1} \leq \varepsilon,$$

concluding the claim. □

Remark D.10 (Dependency of β on ν^*). To make the comparison between the constants in Corollary D.21 and Corollary D.22 easier, we note that

$$2\nu^* + 2 \geq \nu^* + \sqrt{\nu^* + 2},$$

simplifying the term in $\beta(\varepsilon)$ in Corollary D.22 to

$$(2\nu^* + 2)(\nu^* + 1) = 2(\nu^* + 1)^2 > (\nu^* + 1)\left(\nu^* + \sqrt{\nu^{*2} + 2}\right),$$

corresponding to the term in Corollary D.21 with an additional factor of $\nu^* + 1$.

Under the assumption of strong duality, we translate the results of to convergence guarantees in terms of the function optimality gap and the constraint violation.

D.3.2 Sample Complexity Analysis

Corollary D.11 (IPPPM+SM, Non-Smooth, Exact Penalty)**.** *Under the assumptions of Theorem D.8, running Algorithm 2 with the* ***exact penalty*** $\mathcal{L}(\cdot) \coloneqq [\cdot]_+$, $\hat{\rho} \coloneqq 2\rho_\varphi$ *and the* ***sub-gradient method (SM)*** *as an inner solver for*

$$N \geq \frac{3\beta(\nu^*)\rho\mathcal{D}_\Lambda^2}{\mu_c^2}\frac{\beta(\nu^*)}{\varepsilon} \cdot \log\left(\frac{3\beta(\nu^*)[\Delta_0 + \beta(\nu^*)\mathcal{V}_0]}{\varepsilon}\right)$$

outer iterations yields the guarantee

$$\left|F_1(\theta^{(N)}) - F^*\right| \leq \varepsilon \quad \textit{and} \quad [F_2(\theta^{(N)})]_+ \leq \varepsilon$$

as a special case of (IPPPM-Qu)*. SM has a complexity of*

$$T_{\text{in}}^{\text{SM}}(\varepsilon_{\text{in}}) \geq \frac{12G_\varphi^2\beta(\nu^*)\mathcal{D}_\Lambda^2}{\mu_c^2}\varepsilon^{-2} = \mathcal{O}\big(\varepsilon^{-2}\big)$$

to achieve the desired inner accuracy, where $\Delta_0 \coloneqq F_1(\theta^{(0)}) - F^*$, $\mathcal{V}_0 \coloneqq [F_2(\theta^{(0)})]_+$, $\beta(\nu^*) \coloneqq 2\nu^* + 2$, $\rho_\varphi(\beta) \coloneqq \beta(\nu^*) \cdot \rho$, *and* $G_\varphi(\beta) \coloneqq G + \beta(\nu^*)\rho\mathcal{D}_\Theta + \beta(\nu^*)G$.

The total number of sub-gradient and function evaluations is Consequently given by:

$$T_{\text{tot}}^{\text{n-sm}} = N \cdot T_{\text{in}}^{\text{SM}}(\varepsilon_{\text{in}}) = \widetilde{\mathcal{O}}\big(\varepsilon^{-3}\big).$$

Proof. We invoke Lemma D.19 and obtain that each problem (IPPPM) is ρ_φ-weakly convex, with $\rho_\varphi = \beta\rho$ and has uniformly bounded gradients by $G_\varphi \coloneqq G + \rho_\varphi\mathcal{D}_\Theta + \beta G$. Thus, we can lift with $\hat{\rho}(\beta) \coloneqq 2\rho_\varphi = \beta(\nu^*)\rho$ and obtain $(\beta(\nu^*) \cdot \rho)$-strongly convex inner problems. Plugging in the constants into Theorem D.8 yields

$$\begin{aligned} N &\geq \frac{3\hat{\rho}(\beta) \cdot \mathcal{D}_\Lambda^2}{2\mu_c^2}\frac{\beta(\nu^*)}{\varepsilon} \cdot \log\left(\frac{3\beta(\nu^*)[\Delta_0 + \beta(\nu^*)\mathcal{V}_0]}{\varepsilon}\right) \\ &= \frac{3\beta(\nu^*)\rho\mathcal{D}_\Lambda^2}{\mu_c^2}\frac{\beta(\nu^*)}{\varepsilon} \cdot \log\left(\frac{3\beta(\nu^*)[\Delta_0 + \beta(\nu^*)\mathcal{V}_0]}{\varepsilon}\right) \end{aligned}$$

for the number of outer iterations. For the inner solver, we use the sub-gradient method (SM) for simplicity with a desired accuracy of

$$\varepsilon_{\text{in}} \leq \frac{\alpha}{2\beta(\nu^*)}\varepsilon \leq \frac{2\mu_c^2}{3\hat{\rho}(\beta) \cdot \mathcal{D}_\Lambda^2}\varepsilon \cdot \frac{1}{2\beta(\nu^*)}\varepsilon = \frac{\mu_c^2}{3\beta(\nu^*)^2\rho\mathcal{D}_\Lambda^2}\varepsilon^2.$$

Invoking Corollary 3.3 in [64] yields

$$\begin{aligned} T_{\text{in}}^{\text{SM}}(\varepsilon_{\text{in}}) &\geq \frac{4G_\varphi(\beta)^2}{\beta(\nu^*)\rho \cdot \varepsilon_{\text{in}}} = \frac{4G_\varphi(\beta)^2}{\beta(\nu^*)\rho} \cdot \frac{3\beta(\nu^*)^2\rho\mathcal{D}_\Lambda^2}{\mu_c^2}\varepsilon^{-2} \\ &= \frac{12\beta(\nu^*)G_\varphi(\beta)^2 \cdot \mathcal{D}_\Lambda^2}{\mu_c^2}\varepsilon^{-2}. \end{aligned}$$

Multiplying both rates concludes the claim. □

Corollary D.12 (IPPPM+GD, Smooth, Quadratic Penalty)**.** *Under the assumptions of Theorem D.6, and assuming that F_1, F_2 are L-smooth, running Algorithm 2 with the* ***quadratic penalty*** $\mathcal{L}(\,\cdot\,) \coloneqq [\,\cdot\,]_+^2$, $\hat{\rho}(\varepsilon) \coloneqq 2\rho_\varphi(\varepsilon)$, *and* $\beta = \beta(\varepsilon) \coloneqq \frac{2(\nu^*+1)^2}{\varepsilon}$ *yields the guarantee*

$$\left|F_1(\theta^{(N)}) - F^*\right| \leq \varepsilon \quad \textit{and} \quad [F_2(\theta^{(N)})]_+ \leq \varepsilon.$$

The total number of gradient evaluations when using the ***gradient descent*** *(GD) method as the inner solver, is given by:*

$$\begin{aligned} T_{\text{tot}}^{\text{sm}} &= \frac{\mathcal{D}_\Lambda^2}{\mu_c^2}\beta(\varepsilon)\cdot\log(3\beta(\varepsilon)[\Delta_0 + \beta(\varepsilon)\mathcal{V}_0])\cdot \\ &\qquad \cdot L_\varphi(\varepsilon)\cdot\log\left(\frac{L_\varphi(\varepsilon)\mathcal{D}_\Theta^2}{\varepsilon_{\text{in}}}\right) \\ &= \widetilde{\mathcal{O}}\big(\varepsilon^{-2}\big), \end{aligned}$$

where $\Delta_0 \coloneqq F_1(\theta^{(0)}) - F^*$, $\mathcal{V}_0 \coloneqq [F_2(\theta^{(0)})]_+^2$, $L_\varphi(\varepsilon) = L + \hat{\rho} + \beta(\varepsilon)\cdot G(\mathcal{D}_\Theta L + G) = \mathcal{O}\big(\varepsilon^{-1}\big)$ *and* $\rho_\varphi(\varepsilon) = \beta(\varepsilon)G\mathcal{D}_\Theta \cdot \min\{\rho, L\} = \mathcal{O}\big(\varepsilon^{-1}\big)$.

Proof. We invoke Lemma D.17 to obtain that each subproblem (IPPPM) is $\rho_\varphi(\varepsilon)$-weakly convex with

$$\rho_\varphi(\varepsilon) = \beta(\varepsilon)\cdot G\mathcal{D}_\Theta\cdot\min\{\rho, L\} = \mathcal{O}\big(\varepsilon^{-1}\big),$$

and L_φ-smooth, where

$$L_\varphi(\varepsilon) = L_\varphi = L + \hat{\rho}(\varepsilon) + \beta(\varepsilon)\cdot G(\mathcal{D}_\Theta L + G) = \mathcal{O}\big(\varepsilon^{-1}\big).$$

We define $\hat{\rho}(\varepsilon) = \hat{\rho} \coloneqq 2\cdot\rho_\varphi(\varepsilon)$ and by invoking Theorem D.9 we get

$$N \geq \frac{\hat{\rho}(\varepsilon)\mathcal{D}_\Lambda^2}{2\mu_c^2}\beta(\varepsilon)\cdot\log(3\beta(\varepsilon)[\Delta_0 + \beta(\varepsilon)\mathcal{V}_0])$$

as the number of outer iterations. To solve the inner $\rho_\varphi(\varepsilon)-$strongly convex, $L_\varphi(\varepsilon)$-smooth problems with an accuracy of

$$\varepsilon_{\text{in}} \leq \frac{\alpha}{2\beta(\varepsilon)} \leq \frac{2\mu_c^2}{3\hat{\rho}(\varepsilon)\mathcal{D}_\Lambda^2}\varepsilon\frac{1}{2\beta(\varepsilon)} = \frac{\mu_c^2}{6\mathcal{D}_\Lambda^2}\cdot\frac{\varepsilon}{\beta(\varepsilon)\cdot\rho_\varphi(\varepsilon)} = \mathcal{O}\big(\varepsilon^3\big)$$

we use the gradient descent method, resulting in an inner complexity of

$$T_{\text{in}}^{\text{GD}}(\varepsilon_{\text{in}}) \geq \frac{L_\varphi(\varepsilon)}{\rho_\varphi(\varepsilon)}\cdot\log\left(\frac{L_\varphi(\varepsilon)\mathcal{D}_\Theta^2}{\varepsilon_{\text{in}}}\right),$$

cf. [99]. Multiplying the two rates gives

$$\begin{aligned} T_{\text{tot}}^{\text{sm}} &\geq N\cdot T_{\text{in}}^{\text{GD}}(\varepsilon_{\text{in}}) \\ &= \frac{2\rho_\varphi(\varepsilon)\mathcal{D}_\Lambda^2}{2\mu_c^2}\beta(\varepsilon)\cdot\log(3\beta(\varepsilon)[\Delta_0 + \beta(\varepsilon)\mathcal{V}_0])\cdot\frac{L_\varphi(\varepsilon)}{\rho_\varphi(\varepsilon)}\cdot\log\left(\frac{L_\varphi(\varepsilon)\mathcal{D}_\Theta^2}{\varepsilon_{\text{in}}}\right) \\ &= \frac{\mathcal{D}_\Lambda^2}{\mu_c^2}\beta(\varepsilon)\cdot\log(3\beta(\varepsilon)[\Delta_0 + \beta(\varepsilon)\mathcal{V}_0])\cdot L_\varphi(\varepsilon)\cdot\log\left(\frac{L_\varphi(\varepsilon)\mathcal{D}_\Theta^2}{\varepsilon_{\text{in}}}\right) \\ &= \mathcal{O}\big(\varepsilon^{-2}\cdot\log(\varepsilon^{-2})\cdot\log(\varepsilon^{-4})\big) = \widetilde{\mathcal{O}}\big(\varepsilon^{-2}\big) \end{aligned}$$

and consequently concludes the claim. □

Corollary D.13 (IPPPM+SM, Non-Smooth, Quadratic Penalty)**.** *Under the assumptions of Theorem D.8 running Algorithm 2 with the* ***quadratic penalty*** $\mathcal{L}(\,\cdot\,) \coloneqq [\,\cdot\,]_+^2$, $\beta(\varepsilon) \coloneqq (\nu^*+1)\frac{\nu^*+\sqrt{\nu^{*2}+2}}{\varepsilon}$, *and* $\hat{\rho}(\varepsilon) \coloneqq 2\cdot\rho_\varphi(\varepsilon)$ *yields the guarantee*

$$\left|F_1(\theta^{(N)}) - F^*\right| \leq \varepsilon \quad \textit{and} \quad [F_2(\theta^{(N)})]_+ \leq \varepsilon.$$

*The total number of sub-gradient evaluations when using the **sub-gradient method** (SM) as an inner solver, is given by:*

$$T_{\text{tot}}^{\text{n-sm}} \geq \widetilde{\mathcal{O}}\big(\varepsilon^{-2}\big) \cdot \mathcal{O}\big(\varepsilon^{-4}\big) = \widetilde{\mathcal{O}}\big(\varepsilon^{-6}\big),$$

where $G_\varphi(\varepsilon) \coloneqq G + \hat{\rho}(\varepsilon) \cdot \mathcal{D}_\Theta + 2\beta(\varepsilon) \cdot \mathcal{D}_\Theta G^2 = \mathcal{O}\big(\varepsilon^{-1}\big)$ *and* $\rho_\varphi(\varepsilon) = \beta(\varepsilon) \cdot G\mathcal{D}_\Theta \cdot \min\{\rho, L\} = \mathcal{O}\big(\varepsilon^{-1}\big)$.

Proof. We invoke Lemma D.17 to obtain that each subproblem (IPPPM) is $\rho_\varphi(\varepsilon)$-weakly convex with

$$\rho_\varphi(\varepsilon) = \beta(\varepsilon) \cdot G\mathcal{D}_\Theta \cdot \min\{\rho, L\} = \mathcal{O}\big(\varepsilon^{-1}\big),$$

and has $G_\varphi(\varepsilon)$ bounded sub-gradients, where

$$G_\varphi(\varepsilon) \coloneqq G + \hat{\rho}(\varepsilon) \cdot \mathcal{D}_\Theta + 2\beta(\varepsilon) \cdot \mathcal{D}_\Theta G^2 = \mathcal{O}\big(\varepsilon^{-1}\big).$$

We define $\hat{\rho}(\varepsilon) \coloneqq 2 \cdot \rho_\varphi(\varepsilon)$ and by invoking Theorem D.9 we get

$$N \geq \frac{\hat{\rho}(\varepsilon)\mathcal{D}_\Lambda^2}{2\mu_c^2}\beta(\varepsilon) \cdot \log(3\beta(\varepsilon)[\Delta_0 + \beta(\varepsilon)\mathcal{V}_0]) = \widetilde{\mathcal{O}}\big(\varepsilon^{-2}\big)$$

as the number of outer iterations. To solve the inner $\rho_\varphi(\varepsilon)-$strongly convex problems with an accuracy of

$$\varepsilon_{\text{in}} \leq \frac{\alpha}{2\beta(\varepsilon)} \leq \frac{2\mu_c^2}{3\hat{\rho}(\varepsilon)\mathcal{D}_\Lambda^2}\varepsilon\frac{1}{2\beta(\varepsilon)} = \frac{\mu_c^2}{6\mathcal{D}_\Lambda^2} \cdot \frac{\varepsilon}{\beta(\varepsilon) \cdot \rho_\varphi(\varepsilon)} = \mathcal{O}\big(\varepsilon^3\big)$$

we use the sub-gradient method (SM), resulting in an inner complexity of

$$\begin{aligned} T_{\text{in}}^{\text{SM}}(\varepsilon_{\text{in}}) &\geq \frac{4G_\varphi(\varepsilon)^2}{\rho_\varphi(\varepsilon) \cdot \varepsilon_{\text{in}}} \geq \frac{4G_\varphi(\varepsilon)^2}{\rho_\varphi(\varepsilon)}\frac{6\mathcal{D}_\Lambda^2}{\mu_c^2} \cdot \frac{\beta(\varepsilon) \cdot \rho_\varphi(\varepsilon)}{\varepsilon} \\ &= \frac{24\mathcal{D}_\Lambda^2}{\mu_c^2} \cdot \frac{G_\varphi(\varepsilon)^2\beta(\varepsilon)}{\varepsilon} = \mathcal{O}\big(\varepsilon^{-4}\big), \end{aligned}$$

cf. Corollary 3.3 in Lan and Zhou [67]. Multiplying the two rates concludes the claim. □

A direct comparison between Corollary D.13 and Corollary D.11 reveals that, in the non-smooth case, the exact penalty function is the natural choice, leading to improved numerical constants since $[\,\cdot\,]_+$ preserves the weak convexity and the upper bound on the sub-gradients without introducing additional constants.

In contrast, in the smooth case, the quadratic penalty function $[\,\cdot\,]_+^2$ has to be used to smooth the penalty term. While this approach ensures differentiability, it slightly worsens the convergence guarantee compared the results in Fatkhullin et al. [46].

D.4 PPPM under Hidden Strong Convexity

In the case of hidden *strong* convexity, we can significantly strengthen the result of Theorem D.6 to a logarithmic sample complexity in ε^{-1} for the number of outer loops.

Theorem D.14 (Hidden Strongly Convex, PPPM, Penalty Function)**.** *Assume* (HC) *is hidden* strongly *convex with modulus* $\mu_H > 0$ *and we have access to an oracle* $\mathcal{A}$ *according to Assumption D.5. For a target precision* $\varepsilon > 0$*, lifting parameter* $\hat{\rho} > \rho$*, and a penalty parameter* $\beta \geq 1$*. With the choice of* $\alpha \coloneqq \frac{\mu_c^2\mu_H}{\hat{\rho}+\mu_c^2\mu_H} \in [0, 1]$ *and for* $N \in \mathbb{N}$*, the last iterate of Algorithm 2 satisfies*

$$P_{\mathcal{L},\beta}^{\mathsf{opt}}(\theta^{(N)}) = F_1(\theta^{(N)}) - F^* + \beta\mathcal{L}(F_2(\theta^{(N)})) \leq \varepsilon$$

after

$$N \geq \frac{\hat{\rho} + \mu_c^2\mu_H}{\mu_c^2\mu_H} \cdot \log\left(\frac{2(\Delta_0 + \beta\mathcal{V}_0)}{\varepsilon}\right)$$

iterations, where $\Delta_0 \coloneqq F_1(\hat{\theta}^{(k)}) - F^*$ *and* $\mathcal{V}_0 \coloneqq \mathcal{L}(F_2(\theta^{(0)}))$*. With an inner precision of* $\varepsilon_{\text{in}} \leq \frac{\alpha\varepsilon}{2}$*, the total rate is in general given by*

$$T_{\text{tot}} \geq N \cdot T_{\text{in}}(\varepsilon_{\text{in}}) = \frac{\hat{\rho} + \mu_c^2\mu_H}{\mu_c^2\mu_H} \cdot \log\left(\frac{2(\Delta_0 + \beta\mathcal{V}_0)}{\varepsilon}\right) \cdot T_{\text{in}}\left(\frac{1}{2}\alpha\varepsilon\right).$$

Proof. Similar to the proof of Theorem 5 in Fatkhullin et al. [46], we can pick α uniformly small to obtain a recursion of the form

$$F_1(\theta^{(k+1)}) - F^* + \beta\mathcal{L}(F_2(\theta^{(k+1)})) \leq (1-\alpha)(F_1(\theta^{(k)}) - F^* + \beta\mathcal{L}(F_2(\theta^{(k)}))) + \varepsilon_{\text{in}}.$$

By unrolling, we obtain

$$P^{\mathsf{opt}}_{\mathcal{L},\beta}(\theta^{(N)}) = F_1(\theta^{(N)}) - F^* + \beta\mathcal{L}(F_2(\theta^{(N)})) \leq (1-\alpha)^N(\Delta_0 + \beta) + \frac{\varepsilon_{\text{in}}}{\alpha}.$$

The claim follows by the choice of $\alpha, \varepsilon_{\text{in}}$ and N. □

D.5 Technical Lemmas

For our penalty approach, we are generally interested in weak convexity of the penalty term, a (generalized) chain rule as well as bounded sub-gradients and L-smoothness respectively. For the chain rule in the smooth case, we use the result of Drusvyatskiy and Paquette [41], and in the non-smooth case we generalize their results to both, the special case of the quadratic penalty $[\,\cdot\,]_+^2$ and the exact penalty $[\,\cdot\,]_+$.

Before we start analyzing the two cases separately, we state a few utility Lemmas.

Lemma D.15 (Lipschitz Continuity, Quadratic Penalty)**.** *Let $h : \mathcal{Y} \to \mathbb{R}, \tilde{\theta} \mapsto [\tilde{\theta}]_+^2$, $\mathcal{Y} \subset \mathbb{R}^m$, $m \in \mathbb{N}$ be the quadratic penalty function over a $\mathcal{D}_\mathcal{Y}$-bounded domain. Then h is $2 \cdot \mathcal{D}_\mathcal{Y}$-Lipschitz continuous.*

Furthermore, if composed with a G_f-Lipschitz continuous function $f : \Theta \to \mathbb{R}^m$, over a $\mathcal{D}_\Theta$-bounded, compact domain $\Theta \subset \mathbb{R}^d$, the resulting function $h \circ f$ is $2 \cdot \mathcal{D}_\Theta \cdot G_f^2$-Lipschitz continuous.

Proof. We first consider only the quadratic norm. Since we assume to have a feasible point for our problem (HC), we can assume that $\mathcal{Y}$ contains the origin. More generally, one would obtain $2\sup_{v\in\mathcal{Y}}\|[v]_+\|$ as a Lipschitz constant.

The penalty function is globally $2 \cdot \mathcal{D}_\mathcal{Y}$-Lipschitz continuous. I.e. for $v, w \in \text{Im}(f)$ we have

$$|h(v) - h(w)| = \left|\|v\|^2 - \|w\|^2\right| = |\langle v + w, v - w\rangle| \leq (\|v\| + \|w\|) \cdot \|v - w\|, \tag{12}$$

which implies $2 \cdot \mathcal{D}_\Theta$-Lipschitz continuity on a bounded domain. The same also holds true if we substitute $\|\cdot\|^2$ by $[\,\cdot\,]_+^2$.

Since f is Lipschitz continuous and Θ is compact and bounded, also the image $\mathcal{Y} \coloneqq \text{Im}(f) \coloneqq f(\Theta)$ is compact and bounded by $G_f \cdot \mathcal{D}_\Theta$. Invoking (12) with $v = f(\theta), w = f(\tilde{\theta}), \theta, \tilde{\theta} \in \Theta$, yields that $h \circ f$ is $(2 \cdot G_f \cdot \mathcal{D}_\Theta \cdot G_f)$-Lipschitz continuous, concluding the claim. □

Lemma D.16 (Weak Convexity of Composition)**.** *Let $f : \Theta \to \mathbb{R}$, with $\Theta \subset \mathbb{R}^d$ bounded and convex, be ρ-weakly convex with a bounded range, and let $h : \mathbb{R} \to \mathbb{R}$ be non-decreasing, convex and G_h-Lipschitz continuous on* $\text{range}(f)$*. Then the composition $g \coloneqq h \circ f : \Theta \to \mathbb{R}$ is $(G_h \cdot \rho)$-weakly convex.*

Proof. To show μ-weak convexity, we need to prove that for all $\theta, \tilde{\theta} \in \Theta$ and all $t \in [0, 1]$ the inequality

$$g(t\theta + (1-t)\tilde{\theta}) \leq tg(\theta) + (1-t)g(\tilde{\theta}) + \frac{\mu}{2}t(1-t)\|\theta - \tilde{\theta}\|^2$$

holds. Since by assumption f is ρ-WC, we know that the function $\tilde{f}(\theta) \coloneqq f(\theta) + \frac{\rho}{2}\|\theta\|^2$ is convex. Let $\theta, \tilde{\theta} \in \Theta$ and $t \in [0, 1]$ be arbitrary but fixed. By convexity of $\tilde{f}$, we obtain

$$f(t\theta + (1-t)\tilde{\theta}) + \frac{\rho}{2}\|t\theta + (1-t)\tilde{\theta}\|^2 \leq t\Big(f(\theta) + \frac{\rho}{2}\|\theta\|^2\Big) + (1-t)\Big(f(\tilde{\theta}) + \frac{\rho}{2}\|\tilde{\theta}\|^2\Big).$$

Since the Euclidean norm is 2-strongly convex, we have

$$\|t\theta + (1-t)\tilde{\theta}\|^2 = t\|\theta\|^2 + (1-t)\|\tilde{\theta}\|^2 - t(1-t)\|\theta - \tilde{\theta}\|^2.$$

and thus the left-hand side expands to

$$f(t\theta + (1-t)\tilde{\theta}) + \frac{\rho}{2}\|t\theta + (1-t)\tilde{\theta}\|^2$$
$$= f(t\theta + (1-t)\tilde{\theta}) + \frac{\rho}{2}\Big[t\|\theta\|^2 + (1-t)\|\tilde{\theta}\|^2 - t(1-t)\|\theta - \tilde{\theta}\|^2\Big].$$

Rearranging yields

$$f(t\theta + (1-t)\tilde{\theta}) \leq tf(\theta) + (1-t)f(\tilde{\theta}) + \underbrace{\frac{\rho}{2}(1-t)t\|\theta - \tilde{\theta}\|^2}_{=:\delta \geq 0}.$$

By inserting the above into h we obtain

$$\begin{aligned} g\Big(t\theta + (1-t)\tilde{\theta}\Big) &= h\Big(f(t\theta + (1-t)\tilde{\theta})\Big) \\ &\overset{(i)}{\leq} h\Big(tf(\theta) + (1-t)f(\tilde{\theta}) + \delta\Big) \\ &\overset{(ii)}{\leq} h\Big(tf(\theta) + (1-t)f(\tilde{\theta})\Big) + G_h \cdot \delta \\ &\overset{(iii)}{\leq} th(f(\theta)) + (1-t)h\Big(f(\tilde{\theta})\Big) + G_h \cdot \delta \\ &= tg(\theta) + (1-t)g(\tilde{\theta}) + \frac{G_h \cdot \rho}{2}(1-t)t\|\theta - \tilde{\theta}\|^2 \end{aligned}$$

by applying in (i) that h is non-decreasing, in (ii) the Lipschitz continuity of h, and in (iii) the convexity of h. This concludes the claim. □

D.5.1 Smooth Case: Quadratic Penalty

We show that the penalty term is weakly convex, satisfies the chain rule and we give a uniform upper bound for the gradients.

Lemma D.17 (Smooth, Quadratic Penalty)**.** *Let $\Theta \subset \mathbb{R}^d$ be a $\mathcal{D}_\Theta$-bounded, compact domain, and $h : \mathbb{R} \to \mathbb{R}, \theta \mapsto [\theta]_+^2$ be the **quadratic penalty** function. We assume that $f : \Theta \to \mathbb{R}$ is L-smooth, ρ-weakly convex and has G_f uniformly bounded gradients. Then the composition $g := h \circ f : \Theta \to \mathbb{R}$ satisfies:*

- *(Weak Convexity) The penalty term g is $(2G_f\mathcal{D}_\Theta \cdot \min\{\rho, L\})$-weakly convex.*
- *(Chain Rule) For all $\theta \in \Theta$ we have*
$$\nabla g(\theta) = h'(f(\theta)) \cdot \nabla f(\theta) = 2[f(\theta)]_+ \cdot \nabla f(\theta).$$
- *(Bounded Gradients) The gradients of g are uniformly bounded by $G_g := 2\mathcal{D}_\Theta G_f^2$.*
- *(L_g-smoothness) The penalty term g is L_g-smooth, with $L_g := 2(F_{\mathsf{UB}}L + G_f^2)$,*

where $\max_{\theta\in\Theta}|f(\theta)| =: F_{\mathsf{UB}} < \infty$ is an upper bound.

Proof. The weak convexity is a consequence of Lemma D.16 and Lemma 4.2 in Drusvyatskiy and Paquette [41]. We apply Lemma D.15 to obtain that g is Lipschitz continuous. The third claim follows by Theorem 3.1 in Drusvyatskiy and Paquette [41]. To show the L_g-smoothness, let $\theta, \tilde{\theta} \in \Theta$. We have

$$\begin{aligned} \Big\|\nabla g(\theta) - \nabla g(\tilde{\theta})\Big\| &\leq 2 \cdot \Big\|[f(\theta)]_+\nabla f(\theta) - [f(\theta)]_+\nabla f(\tilde{\theta}) + [f(\theta)]_+\nabla f(\tilde{\theta}) - [f(\tilde{\theta})]_+\nabla f(\tilde{\theta})\Big\| \\ &\leq 2 \cdot \|[f(\theta)]_+(\nabla f(\theta) - \nabla f(\tilde{\theta}))\| + 2 \cdot \|([f(\theta)]_+ - [f(\tilde{\theta})]_+)\nabla f(\tilde{\theta})\| \\ &\leq 2 \cdot F_{\mathsf{UB}}L\|\theta - \tilde{\theta}\| + 2 \cdot G_f G_f\|\theta - \tilde{\theta}\| \\ &= 2(F_{\mathsf{UB}}L + G_f^2) \cdot \|\theta - \tilde{\theta}\|, \end{aligned}$$

since for all $\theta \in \Theta$ we have $[f(\theta)]_+ \leq |f(\theta)| \leq \max_{\theta\in\Theta}|f(\theta)| =: F_{\mathsf{UB}} < \infty$. This concludes the claim. □

D.5.2 Non-smooth Case: Exact and Quadratic Penalty

We can generalize the result above to the case of non-smooth f and h being either the quadratic $[\cdot]_+^2$ or exact penalty $[\cdot]_+$.

Lemma D.18 (Non-Smooth, Quadratic Penalty)**.** *Let $f : \Theta \to \mathbb{R}$ be a ρ-weakly convex, (potentially) non-smooth, G_f-Lipschitz continuous function over a $\mathcal{D}_\Theta$-bounded domain $\Theta \subset \mathbb{R}^d$. Furthermore, let $h : \mathbb{R} \to \mathbb{R}, \theta \mapsto [\theta]_+^2$ be the **quadratic penalty** function. Then the composition $g \coloneqq h \circ f$ satisfies:*

- *(Weak Convexity) The penalty term g is $(2G_f\mathcal{D}_\Theta \cdot \rho)$-weakly convex.*
- *(Generalized Chain Rule) For all $\theta \in \Theta$ we have*

$$\partial g(\theta) = 2[f(\theta)]_+ \cdot \partial f(\theta). \tag{13}$$

- *(Bounded Sub-gradients) The sub-gradients of g are uniformly bounded by $G_g \coloneqq 2\mathcal{D}_\Theta G_f^2$.*

Proof. By Lemma D.15 we know that h is $(2G_f\mathcal{D}_\Theta)$—Lipschitz continuous and the third claim is proved. Consequently, the first claim follows by applying Lemma D.16.

Since g is weakly convex, it is *regular* by Proposition 4.4 in Vial [119], according to Definition 3 in Li et al. [70] of regularity, which is by Corollary 8.11 in Rockafellar and Wets [109] equivalent to Definition 6.24 and Corollary 6.29 in Rockafellar and Wets [109] of regularity. Applying the generalized chain rule for sub-gradients, Theorem 10.49 of Rockafellar and Wets [109] and using the fact that h is continuously differentiable concludes the claim. □

Lemma D.19 (Non-Smooth, Exact Penalty)**.** *Let $f : \Theta \to \mathbb{R}$ be a ρ-weakly convex, (potentially) non-smooth, G_f-Lipschitz continuous function over a $\mathcal{D}_\Theta$-bounded domain $\Theta \subset \mathbb{R}^d$. Furthermore, let $h : \mathbb{R} \to \mathbb{R}, \theta \mapsto [\theta]_+$ be the **exact penalty** function. Then the composition $g \coloneqq h \circ f$ satisfies:*

- *(Weak Convexity) The penalty term g is ρ-weakly convex.*
- *(Generalized Chain Rule) For all $\theta \in \Theta$ we have*

$$\partial g(\theta) = \begin{cases} \partial f(\theta), & f(\theta) > 0, \\ [0,1] \cdot \partial f(\theta), & f(\theta) = 0, \\ \{0\}, & f(\theta) < 0, \end{cases} \tag{14}$$

  with a Minkowski product in the second case.

- *(Bounded Sub-gradients) The sub-gradients of g are uniformly bounded by $G_g \coloneqq G_f$.*

Proof. If f is G_f-Lipschitz, then trivially also $h \circ f$ is G_f-Lipschitz. Additionally, $[\cdot]_+$ is non-decreasing, convex, and 1-Lipschitz. Thus, the first claim follows by applying Lemma D.16. With the same argument as above, we can apply the generalized chain rule for sub-gradients, Thm. 10.49 in Rockafellar and Wets [109], and obtain

$$\partial_\theta g(\theta) = \bigcup_{y \in \partial h(f(\theta))} \partial(yf)(\theta),$$

with $\theta \in \Theta$. To obtain (14), we use the known sub-differential of $[\theta]_+$:

$$\partial_\theta [\theta]_+ = \begin{cases} \{1\}, & \theta > 0, \\ [0,1], & \theta = 0, \\ \{0\}, & \theta < 0. \end{cases}$$

Thus, at $f(\theta) = 0$, we obtain a convex combination of sub-gradients $g \in \partial f(\theta)$ and weights $\alpha \in [0,1]$. The result follows. □

D.6 Strong Duality: Translation of Convergence Guarantees

The challenge in Section 5.1 is the translation of the guarantee on $\mathbb{E}\left[P^{\mathsf{opt}}_{\mathcal{L},\beta}(\,\cdot\,)\right]$ to a guarantee on the constraint violation, namely $\mathbb{E}[[F_2(\,\cdot\,)]_+]$. The following results show how to perform this translation under strong duality for both, the exact and the quadratic penalty function. For simplicity, we formulate the following corollaries in the deterministic case. Taking expectation yields the desired results.

Lemma D.20 (Upper Bound Constraint Violation)**.** *Let θ^* be the optimal solution of* (CME) *and assume that strong duality holds, i.e. there exists $\nu^* \geq 0$ such that $F^* = d(\nu^*)$. Then we have for all $\theta \in \Theta$ the inequality*

$$-\nu^*[F_2(\theta)]_+ \leq F_1(\theta) - F^*.$$

In particular, for all $\theta \in \Theta$ and any $\beta \geq 2 \cdot \nu^*$, we have

$$0 \leq F_1(\theta) - F^* + \beta[F_2(\theta)]_+ = P^{\mathsf{opt}}_{1,\beta}(\theta).$$

Proof. Lemma 3.1.21 in Nesterov [98]. □

Corollary D.21 (Constraint Violation, Exact Penalty)**.** *Assume that for* (CME) *strong duality holds with a Lagrangian multiplier $\nu^* \geq 0$, and let $\theta \in \Theta$ be a point such that $\mathbb{E}\left[P^{\mathsf{opt}}_{1,\beta}(\theta)\right] \leq \varepsilon$ with the exact penalty and $\beta > \nu^*$. Then we have the following translation for the constraint violation:*

$$0 \leq \mathbb{E}[[F_2(\theta)]_+] \leq \frac{\mathbb{E}\left[P^{\mathsf{opt}}_{1,\beta}(\theta)\right]}{\beta - \nu^*} \leq \frac{\varepsilon}{\beta - \nu^*}.$$

Proof. We invoke Lemma D.20 and obtain as a trivial consequence:

$$-\nu^*[F_2(\theta)]_+ \leq F_1(\theta) - F^* + \beta[F_2(\theta)]_+ - \beta[F_2(\theta)]_+.$$

Taking expectation concludes the claim. □

Corollary D.22 (Constraint Violation, Quadratic Penalty)**.** *Assume that for* (CME) *strong duality holds with a Lagrangian multiplier $\nu^* \geq 0$, and let $\theta \in \Theta$ be a point such that $\mathbb{E}\left[P^{\mathsf{opt}}_{2,\beta}(\theta)\right] \leq \varepsilon$ with the quadratic penalty. Then we have the following translation for the constraint violation:*

$$\mathbb{E}[[F_2(\theta)]_+] \leq \frac{\nu^* + \sqrt{{\nu^*}^2 + 2\beta\varepsilon}}{\beta}.$$

Proof. We invoke Lemma D.20 and obtain

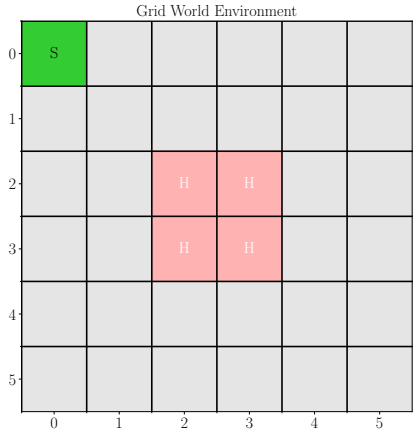
$$\varepsilon \geq \mathbb{E}\left[P^{\mathsf{opt}}_{2,\beta}(\theta)\right] = \mathbb{E}\left[F_1(\theta) - F^* + \beta[F_2(\theta)]_+^2\right] \overset{\text{Jensen}}{\geq} -\nu^*\mathbb{E}[[F_2(\theta)]_+] + \beta\left(\mathbb{E}[[F_2(\theta)]_+]\right)^2.$$

Solving

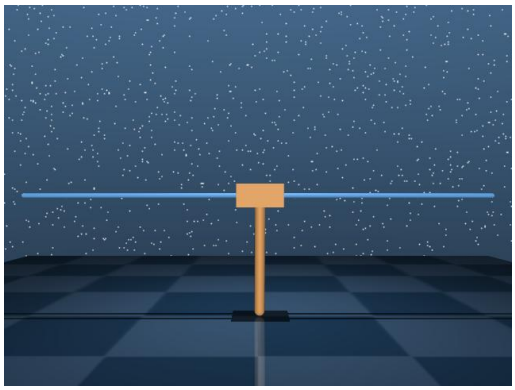
$$s(\beta \cdot s - \nu^*) - \varepsilon \leq 0, \text{ s.t. } s \geq 0$$

concludes the claim. □

E Experimental Details & Ablation Studies



(a) Visualization of the *Frozen Lake* grid-world environment used in Section 6.1 to test and ablate our Algorithm 1. "S" refers to the starting position of the agent, "H" to a (penalized) unsafe hole and " " to a safe tile.



(b) Visualization of the *SafeCartpole* [42] which is an extension of the Google DeepMind Control suite [117, 114]. The pendulum starts in the resting downwards position and the goal is to safely explore the dynamics of the system without exceeding an imposed slider safety limits on the left and right.

Figure 5: Overview of the grid-world with discrete state-action space (left) and the safe cartpole exploration with continuous state-action spaces (right).

E.1 Grid World: Linear Performance Constraint

In this ablation study, we perform a Maximum Entropy exploration under a linear performance constraint, i.e. (CME) of the form

$$\begin{aligned} &\max_{\theta \in \Theta} \mathcal{H}(\lambda^{\pi_\theta}) \\ &\text{s.t. } \langle c, \lambda^{\pi_\theta} \rangle \leq 0 \end{aligned}$$

where the cost function is of the form

$$c(s) = \begin{cases} 50, & s = \texttt{Hole} \\ -0.001, & \text{else} \end{cases}, \tag{$\mathcal{R}_{\text{lin}}$}$$

i.e. penalizing the four holes in the center of the environment Figure 5a. We ablate the penalty parameter $\beta \in \{5 \cdot 10^{-4}, 0.001, 0.005, 0.01\}$, the batch sizes $B \in \{8, 24, 72\}$, and the step sizes $\eta \in \{0.001, 0.01, 0.1\}$. We want to highlight that episodes terminate upon reaching the bottom-right tile. Consequently, the corresponding terminal state has an estimated occupancy $\widehat{\lambda}$ of almost zero for all methods, as it does not contribute informative state-action visitation beyond termination.

The full ablation results of the objective are shown in Figure 7, for the constraint in Figure 8 respectively.

Penalty Parameter β. Since our analysis requires $\beta \sim \varepsilon^{-1}$, we validate empirically that this is actually not required in practice. While bigger β might be indeed necessary to balance the tradeoff between improvement on the objective function and the constraint violation, cf. Figure 8, we ablate in Appendix E.1.2 that the order of magnitude of β rather depends on the scale of the safety constraint and does not necessarily have to be huge in practice. Checkpoint convergence results for both the grid world and SafeCartpole are reported in Table 1 in the main text.

Batch Size B. While the penalty parameter β plays a significant role ensuring constraint violation, the batch size is elementary for the objective function. Larger B becomes more important if one wants to increase the step size to speed-up the training process, cf. Figure 7. Empirically, we cannot observe that batches of $B \sim \varepsilon^{-4}$ are necessary, indicating that our estimates are most likely too conservative.

SGD Step Size η. The step size plays a crucial role, in particular to ensure convergence in the objective. In particular the interplay between the batch size B and η suggests, that both of them have to be chosen proportionally.

Implementation Details Primal-Dual [141]. We implement the primal-dual method of Ying et al. [141] as a baseline and perform a hyperparameter search for the primal and dual learning rate $\alpha_\theta, \alpha_\mu$ and the momentum term α_t in the notation of Ying et al. [141] Algorithm 1, for parameters $\alpha_\theta \in \{0.001, 0.005, 0.01, 0.05, 0.1, 0.5, 1\}$, $\alpha_\mu \in \{0.001, 0.005, 0.01, 0.05, 0.1, 0.5, 1\}$, and $\alpha_t \in \{0.001, 0.005, 0.01, 0.03, 0.05, 0.1, 1\}$. We also ablate the batch size $B \in \{8, 24, 72\}$, as for our PGP-method and pick the best configuration ($\alpha_\theta = 0.01$, $\alpha_\mu = 0.001$, $\alpha = 1$, i.e. no momentum) displayed in Figure 1. Interestingly, also for the primal-dual method, the batch size does not significantly change the performance, which is why we focus in Figure 1 on $B = 8$ for the sake of comparability. In our experiments, we could not see an improvement using their variance reduction momentum scheme: the resulting policies exhibited the oscillating constraint violation behavior and the best performance was achieved with $\alpha_t \equiv 1, t \in [T]$.

E.1.1 Unconstrained Baseline

For comparison, we also run the unconstrained baseline where we maximize entropy without any safety constraints. In particular, we run our penalty algorithm with $\beta = 0.0$ which corresponds to running unconstrained SGD to maximize the entropy and evaluate the constraint violation with respect to ($\mathcal{R}_{\text{lin}}$). The results are shown in Figure 6.

E.1.2 Lower Penalty for Holes, Increased β required

As already suspected in Section 6.1, we find that choosing the penalty parameter β does not have such a strong dependency on $\sim \varepsilon^{-1}$ but rather inversely proportionally depends on the scale of the cost function. To test this hypothesis, we modify the cost function ($\mathcal{R}_{\text{lin}}$) to a relaxed version with a -2 penalty for the Holes, i.e.

$$c(s) = \begin{cases} 2, & s = \texttt{Hole} \\ -0.001, & \text{else} \end{cases}, \qquad (\mathcal{R}_{\text{lin,relax}})$$

as an alternative. As mentioned in the analysis of Algorithm 1 in Section 6.1, we argue that the choice of β in practice does not necessarily scale with $\sim \varepsilon^{-1}$, but should be rather chosen depending on the scale of the reward, i.e. entropy maximization, and the order of magnitude of the constraint, in order to balance the constraint violation and the reward maximization. In particular, if we adjust the safety formulation to ($\mathcal{R}_{\text{lin,relax}}$), i.e. a lower penalty for acting unsafe, we require the penalty parameter β to be larger to achieve similar results, as shown in Figure 10 and Figure 11, respectively, as we expected. Compared to the previous run, also in this case the step size $\eta = 0.1$ seems to be too big to either converge or deliver competitive results.

E.2 Grid World: Reference Trajectory Constraint – Norm Constraint

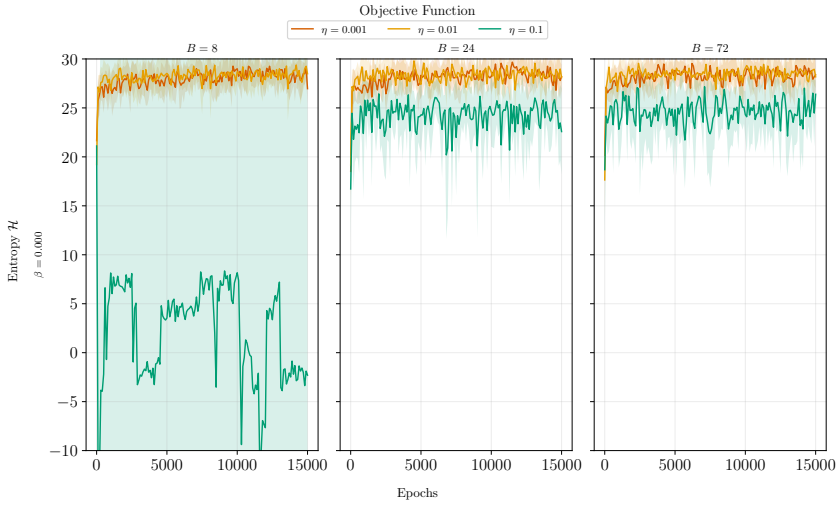
In order to showcase our method's capability of performing maximum entropy exploration beyond linear constraints, we focus on a safe exploration task under imitation learning constraints. The reference policy π_{ref} is, in the notation of Figure 5a, first going all the way to the right and then down until it terminates in the bottom right tile. We encode the constrained problem with an ℓ_2-norm constraint of the form

$$\mathcal{R}_{\ell_2}(\lambda^\pi) := \|\lambda^\pi - \lambda^{\pi_{\text{ref}}}\|_2 \leq b, \qquad (\mathcal{R}_{\ell_2})$$

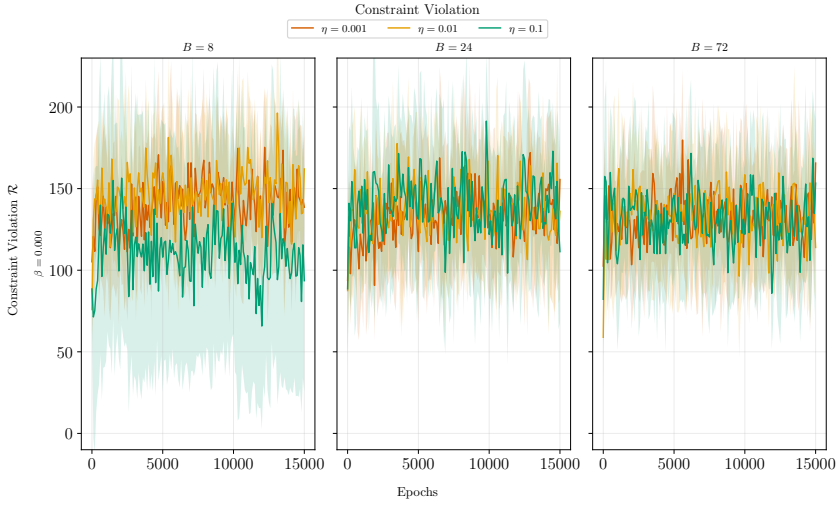
for a constraint violation budget $b \in \mathbb{R}_{>0}$.

We ablate different values of $b \in \{0.0, 0.2, 0.5, 1.0\}$ and run our penalty method with $\beta = 0.5$ and a batch size of $B = 8$.

Results. The final policies for each b are visualized in Figure 12 and the corresponding training progress in Figure 13. One can clearly see that as the shift b increases, the constraint becomes looser; thus the policy can deviate further from the reference policy and obtain a higher entropy value.



(a) Unconstrained **entropy objective**.



(b) Unconstrained **constraint violation**.

Figure 6: Ablation study of **unconstrained entropy maximization** on the environment Figure 5a with the (not considered) safety constraint ($\mathcal{R}_{\text{lin}}$). We ablate different batch sizes $B \in \{8, 24, 72\}$ and step sizes $\eta \in \{0.001, 0.01, 0.1\}$. Each curve shows the mean and one standard deviation, i.e. $\mu \pm \sigma$, for 10 different random seeds.

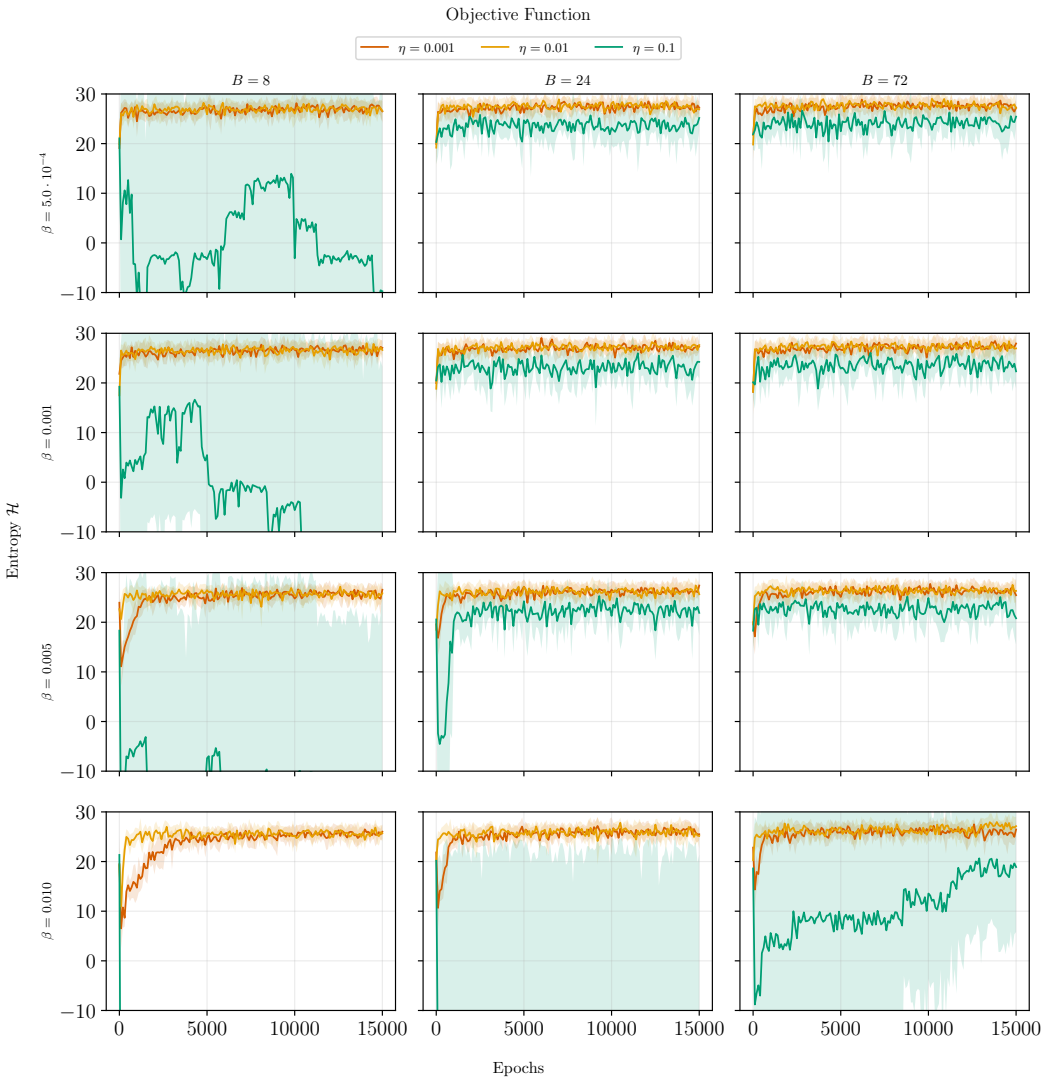


Figure 7: **Ablation study** of the **entropy function value** of our Algorithm 1 under linear constraints ($\mathcal{R}_{\text{lin}}$) running on the environment Figure 5a. We ablate different batch sizes $B \in \{8, 24, 72\}$, penalty parameters $\beta \in \{5 \cdot 10^{-4}, 0.001, 0.005, 0.01\}$ and different step sizes $\eta \in \{0.001, 0.01, 0.1\}$ of Stochastic Gradient Descent. Each curve shows the mean and one standard deviation of the entropy, i.e. $\mu \pm \sigma$, for 10 different random seeds.

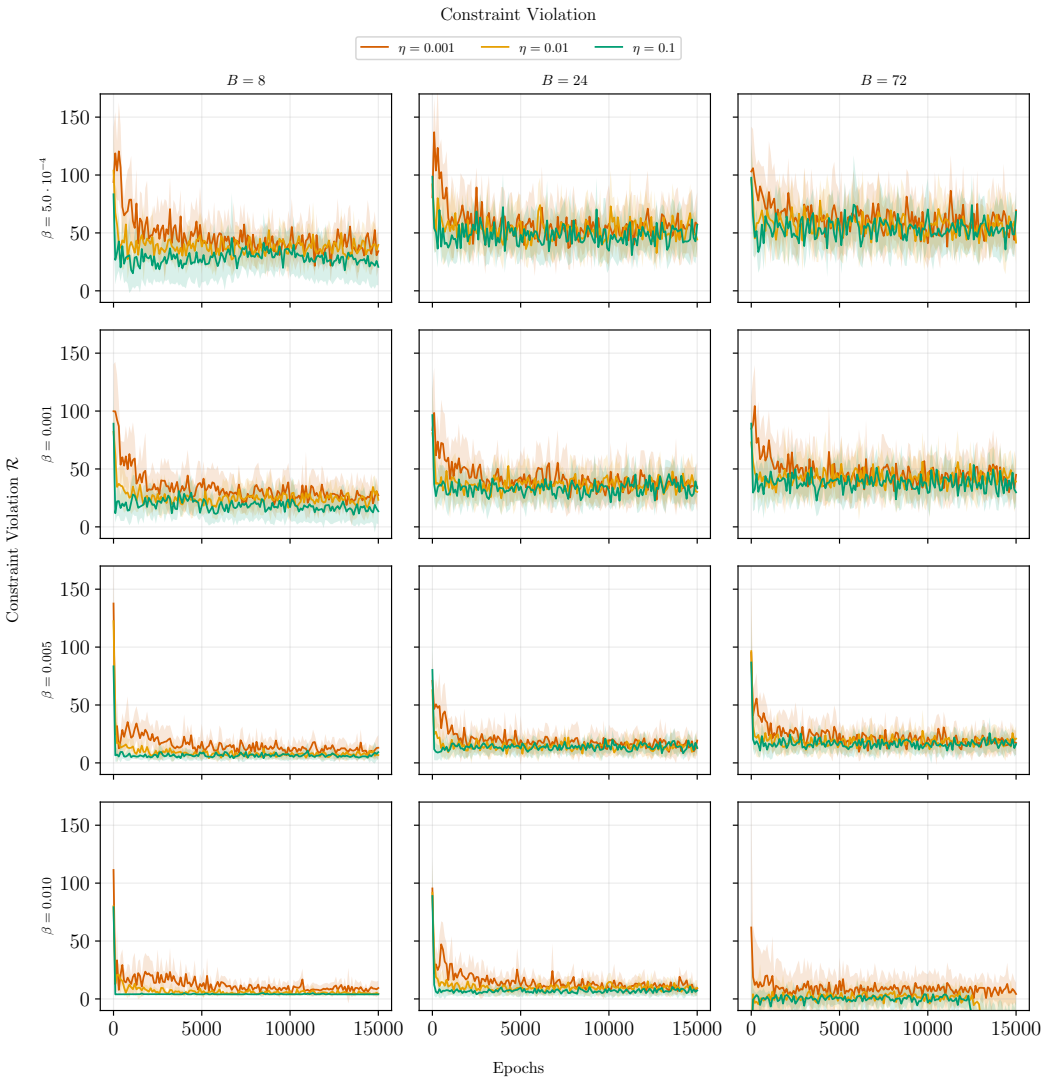
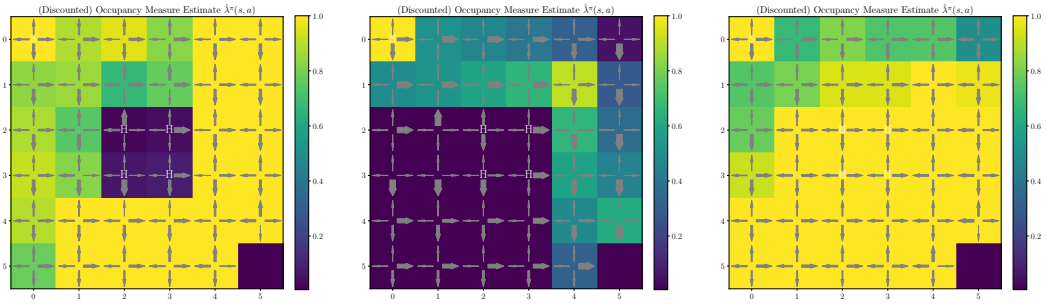


Figure 8: **Ablation study** of the **(linear) constraint function value** Equation ($\mathcal{R}_{\text{lin}}$) for our Algorithm 1 running on the environment Figure 5a. We ablate different batch sizes $B \in \{8, 24, 72\}$, penalty parameters $\beta \in \{5 \cdot 10^{-4}, 0.001, 0.005, 0.01\}$ and different step sizes $\eta \in \{0.001, 0.01, 0.1\}$ of Stochastic Gradient Descent. Each curve shows the mean and one standard deviation of the linear constraint function, i.e. $\mu \pm \sigma$, for 10 different random seeds.



(a) Final policy $\pi_{\theta^{(N)}}$ of executing Penalty Policy Gradient Method in Algorithm 1 with $\beta = 0.005$ and the constraint ($\mathcal{R}_{\text{lin}}$).

(b) Final policy $\pi_{\theta^{(N)}}$ of executing the primal-dual method of Ying et al. [141] with $\alpha_\theta = 0.001, \alpha_t \equiv 1$ (notation of their Algorithm 1) and the constraint ($\mathcal{R}_{\text{lin}}$).

(c) Final policy $\pi_{\theta^{(N)}}$ after running (unconstrained) SGD on (CME), i.e. Algorithm 1 with $\beta = 0.0$.

Figure 9: Policies of Figure 1: Comparison of the estimate of the state-action occupancy measure $\hat{\lambda}^{\pi^{(N)}}$ of the last iterate policies obtained by the Penalty Policy Gradient Method Algorithm 1, the primal-dual method of Ying et al. [141], and running unconstrained SGD to maximize the entropy (without a safety constraint). For PGP and unconstrained SGD, we use a step-size of $\eta = 0.01$ and in all of the three cases we estimate the state-action occupancy measure via (λ_H^{MC}-Est) using a batch size of $B = 8$. One can clearly observe that the penalized version respects the unsafe, penalized tiles in the center, while simply running SGD to maximize the entropy is not safe. The primal-dual algorithm exhibits constraint oscillations and yields a weaker performance, i.e. not fully exploring the space.

E.3 Continuous State-Action Space

We first discuss details on the estimation of the state-action occupancy measure and then provide details on the *PointMass* and the *SafeCartpole* experiment.

E.3.1 Occupancy Measure Estimate

When scaling to continuous state-action spaces in constrained maximum entropy exploration, or more generally in convex RL, reliably estimating the occupancy measure (λ_H^{MC}-Est) is the main bottleneck. In our case, we use and implement the state-of-the-art technique proposed by Barakat et al. [12] relying on Maximum Likelihood Estimators (MLE) for learning the approximation $\widehat{\lambda}$ compared to the Mean Squared Error, which previous techniques have been proposing. Compared to our formulation of $\lambda^\pi : \mathcal{S} \times \mathcal{A} \to [0,1]$ as a function of the state and action space, Barakat et al. [12] rewrite λ^π of a policy π in the form

$$\lambda^\pi(s,a) \coloneqq d^\pi(s) \cdot \pi(a|s),$$

and only approximate the state-occupancy measure d^π with a function approximator.

In particular, at every step of the algorithm, we collect a batch of $N_{\widehat{\lambda}} \in \mathbb{N}$ samples $B_{\widehat{\lambda}} \coloneqq \{s_i, a_i\}_{i \in [N_{\widehat{\lambda}}]}$, consisting of potentially several rollouts and then approximate d^π with a parametric class of probability distributions

$$\text{GMM}(K) \coloneqq \Bigg\{ p_{\mathbf{w}} \coloneqq \sum_{k=1}^{K} w_k \mathcal{N}(\mathbf{m}_k, \Sigma_k) : \mathbf{w} = (w_k, \mathbf{m}_k, \Sigma_k)_{k \in [K]} \\ \text{s.t. } w_k \geq 0, \sum_{k=1}^{K} w_k = 1, \mathbf{m}_k \in \mathbb{R}^{d_s}, \Sigma_k \in \mathbb{S}_{++}^{d_s}, \Bigg\}$$

in our case Gaussian Mixture Models (GMMs), with $K = 16$. The approximation $\hat{d}^\pi$ is learned by maximizing the MLE loss, or equivalently minimizing a Kullback Leibler divergence [101, 127, 112],

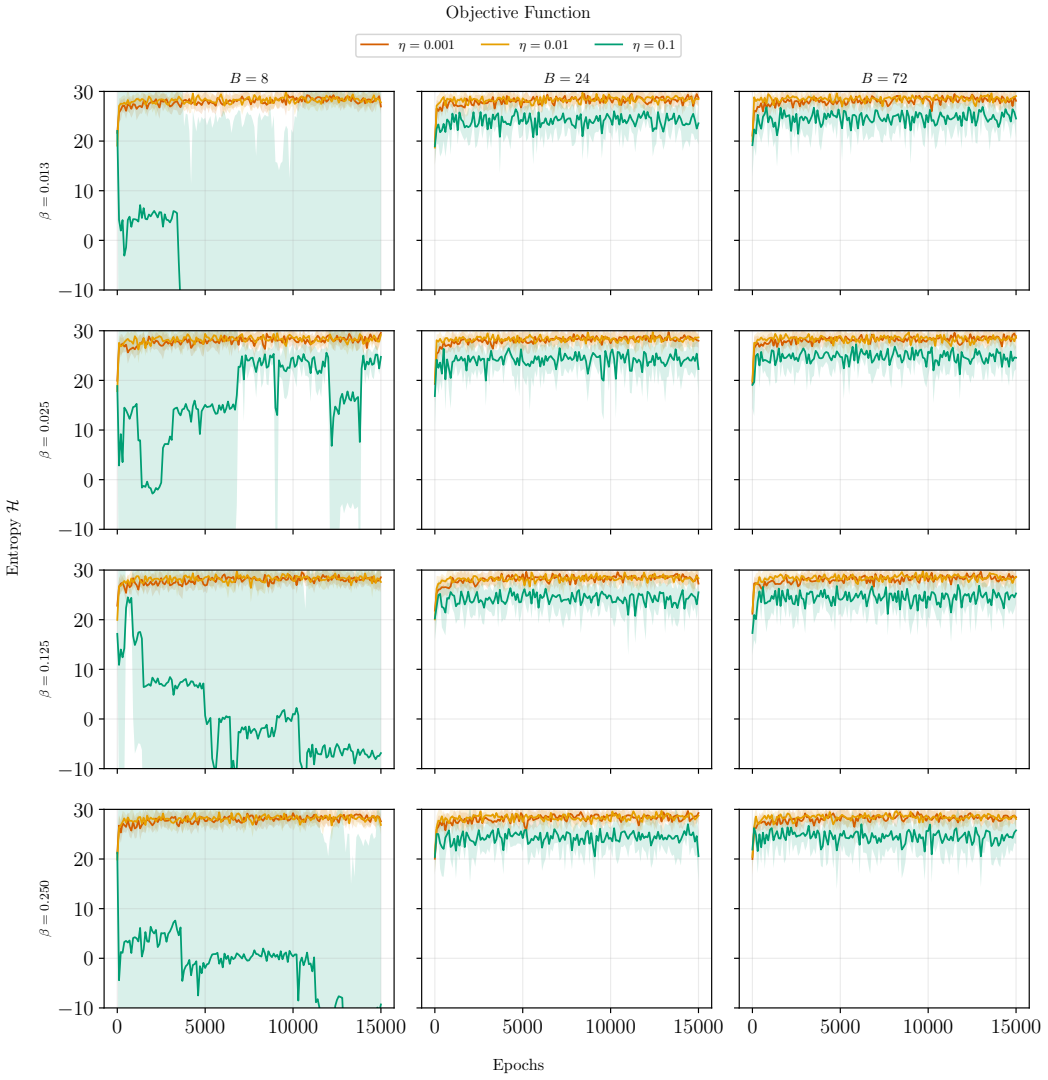


Figure 10: **Ablation study** of the **entropy function value** for our Algorithm 1 under the **relaxed (linear) constraint function value** Equation ($\mathcal{R}_{\text{lin,relax}}$) for our Algorithm 1 running on the environment Figure 5a. We ablate different batch sizes $B \in \{8, 24, 72\}$, **larger penalty parameters** $\beta \in \{0.013, 0.025, 0.125, 0.25\}$ and different step sizes $\eta \in \{0.001, 0.01, 0.1\}$ of Stochastic Gradient Descent. Each curve shows the mean and one standard deviation of the entropy, i.e. $\mu \pm \sigma$, for 10 different random seeds.

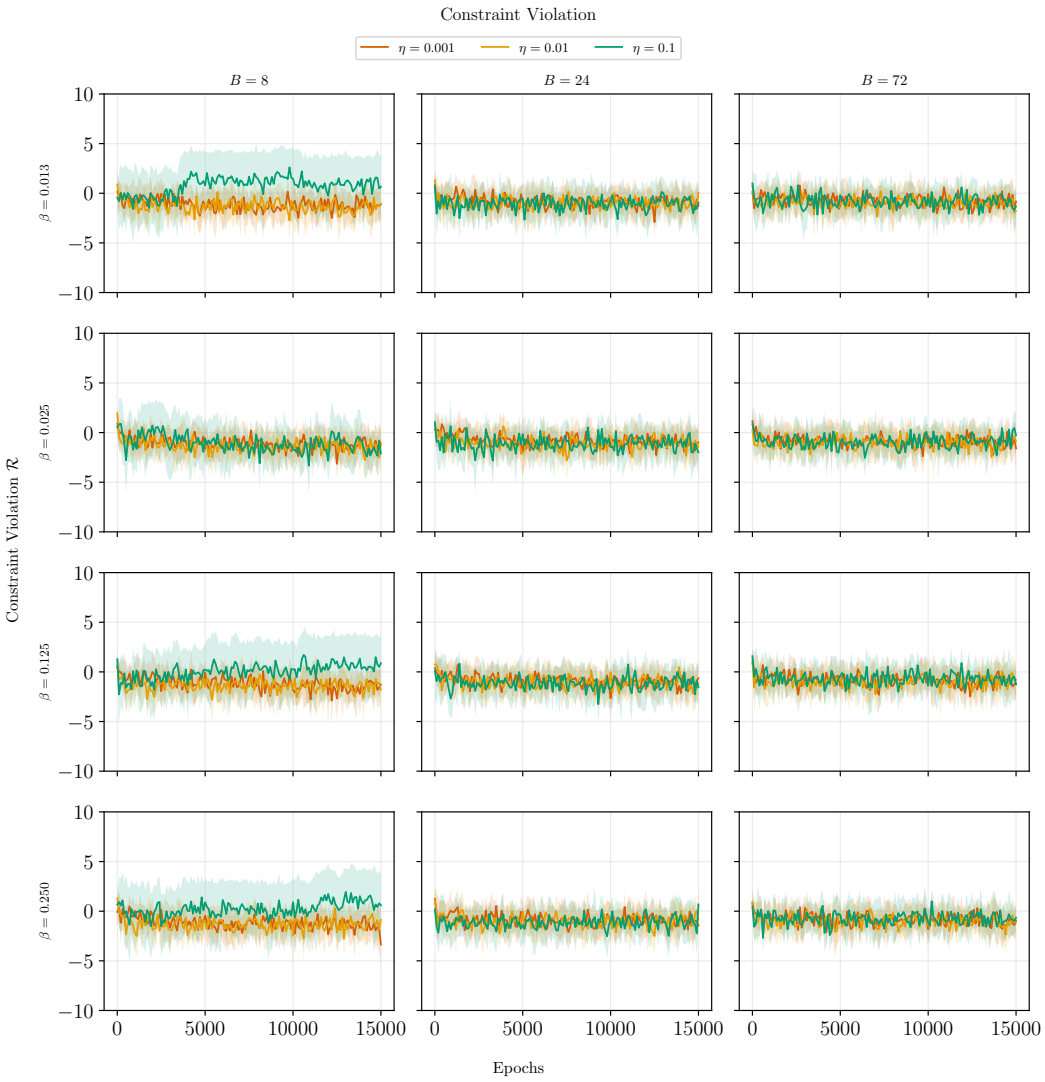


Figure 11: **Ablation study** of the **relaxed (linear) constraint function value** Equation ($\mathcal{R}_{\text{lin,relax}}$) for our Algorithm 1 running on the environment Figure 5a. We ablate different batch sizes $B \in \{8, 24, 72\}$, **larger penalty parameters** $\beta \in \{0.013, 0.025, 0.125, 0.25\}$ and different step sizes $\eta \in \{0.001, 0.01, 0.1\}$ of Stochastic Gradient Descent. Each curve shows the mean and one standard deviation of the linear constraint function, i.e. $\mu \pm \sigma$, for 10 different random seeds.

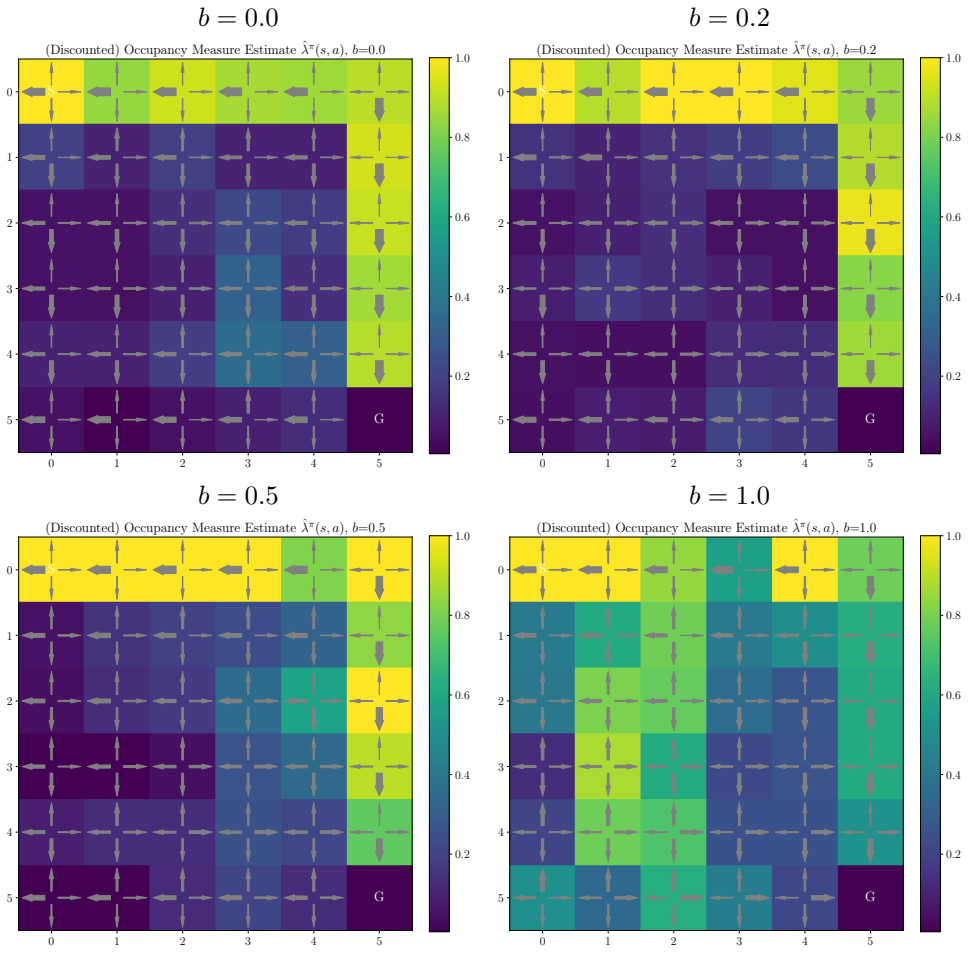


Figure 12: **Final policies** for maximum entropy with norm constraint ($\mathcal{R}_{\ell_2}$) for varying constraint shift parameter b. Each subfigure shows the learned final policy for different shifts $b = 0.0, 0.2, 0.5, 1.0$ (top-left to bottom-right) of the right-hand side of the constraint. Clearly, as b increases, the constraint becomes looser, allowing the learned policy to deviate further from the reference policy.

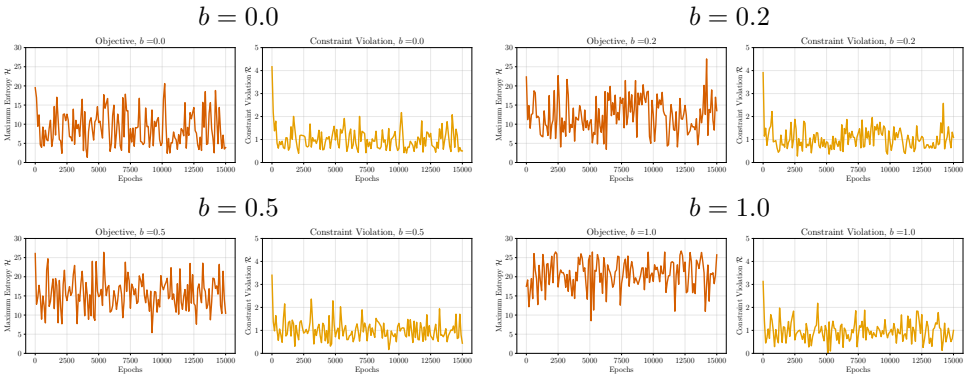


Figure 13: **Training progress** for maximum entropy with norm constraint ($\mathcal{R}_{\ell_2}$) for varying constraint shift parameter b. Each subfigure shows the training curves for $b = 0.0, 0.2, 0.5, 1.0$ (top-left to bottom-right). As b increases, the constraint becomes looser, allowing the learned policy to deviate further from the reference policy.

i.e.

$$\hat{d}^{\pi} \coloneqq p_{\mathbf{w}^*}, \quad p_{\mathbf{w}^*} \in \underset{p_{\mathbf{w}} \in \mathrm{GMM}(K)}{\arg\max} \frac{1}{N_{\widehat{\lambda}}} \sum_{i \in [N_{\widehat{\lambda}}]} \log p_{\mathbf{w}}(s_i). \tag{$\hat{d}^{\pi}$-Est}$$

Following Barakat et al. [12], the estimate $\widehat{\lambda}^{\pi}$ is then given by

$$\widehat{\lambda}^{\pi}(s,a) \coloneqq \hat{d}^{\pi}(s) \cdot \pi(a|s). \tag{λ_H^{MLE}-Est}$$

E.3.2 Imitation Learning Constraint: PointMass Exploration

We provide additional implementation and computational details on the *PointMass* experiment.

State, Action & Observation Space. In the PointMass environment, the state, action and observation space are

$$\mathcal{S} \times \mathcal{A} \times \mathrm{Obs} = \mathbb{R}^2 \times \mathbb{R}^2 \times \mathbb{R}^2.$$

At timestep t, each observation o_t consists of the current x and y coordinate of the point mass. The action a encodes the gain along the x and y-axis respectively.

Objective and Constraint: Definition & Implementation. First, we train a reference policy π_{ref} using the Soft-Actor critic (SAC) algorithm of the *StableBaselines3* package [105]. We fit the reference GMM from stochastic SAC rollouts using the same discounted weighted likelihood as in ($\hat{d}^{\pi}$-Est). On a batch B, the apprenticeship constraint and its pseudo-reward are

$$\widehat{\mathcal{R}}_{\mathrm{KL}} \coloneqq \sum_{i \in B} \bar{w}_i \left(\log \widehat{\lambda}^{\pi}(s_i, a_i) - \log \widehat{\lambda}^{\mathrm{ref}}(s_i, a_i) \right) - \mathcal{R}_{\max},$$

$$r_{\mathrm{KL}}(s_i, a_i) \coloneqq 1 + \log \widehat{\lambda}^{\pi}(s_i, a_i) - \log \widehat{\lambda}^{\mathrm{ref}}(s_i, a_i).$$

The budget $\mathcal{R}_{\max}$ affects the constraint value but, being constant, does not enter its pseudo-reward.

Analysis of Actions. Figure 14 shows that as the constraint budget $\mathcal{R}_{\max}$ increases, the actions of π_θ deviate progressively further from those of π_{ref}. This directly reflects the difficulty of the task: the SAC reference policy consistently directs the agent toward the goal $(0, 0)$, so effective exploration requires substantial deviation from π_{ref}.

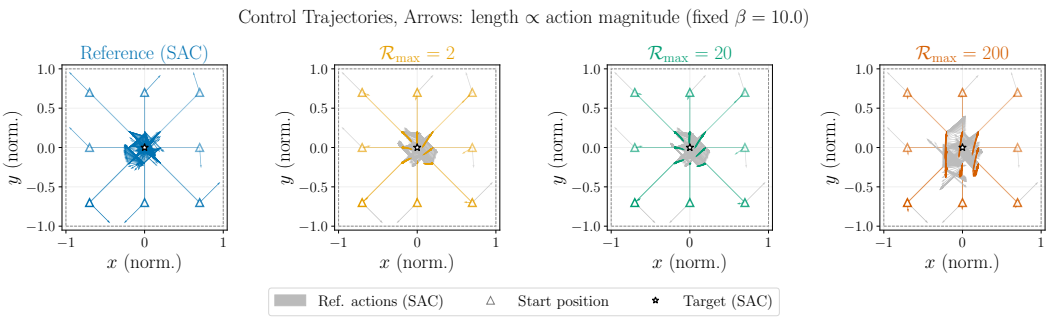


Figure 14: **PointMass: Control Trajectories.** Comparison of actions sampled from $\pi_\theta(\cdot \mid s)$ for the reference SAC policy and PGP-trained policies with $\beta = 10$ and $\mathcal{R}_{\max} \in \{2, 20, 200\}$, evaluated with the same fixed seed across 8 initial positions. Colored arrows indicate the actions selected by each policy at the current state. For PGP policies, the corresponding reference action $\pi_{\mathrm{ref}}(\cdot \mid s)$ is overlaid in gray, illustrating the increasing deviation from the reference as $\mathcal{R}_{\max}$ grows.

E.3.3 Safety Constrained: Cartpole Exploration

We provide implementation and computational details on the *SafeCartpole* experiment.

State, Action & Observation Space. In the SafeCartpole environment, the state, action and observation space are

$$\mathcal{S} \times \mathcal{A} \times \mathrm{Obs} = \mathbb{R}^4 \times \mathbb{R} \times \mathbb{R}^5, \tag{15}$$

where each observation o_t at timestep $t \in [0, T_{\max}]$ consists of

$$o_t = [x_t, \sin(\omega_t), \cos(\omega_t), \frac{\mathrm{d}}{\mathrm{d}t} x_t, \frac{\mathrm{d}}{\mathrm{d}t} \omega_t] \in \mathbb{R}^5, \qquad (o_t\text{-Cartpole})$$

and ω_t is the current angle of the pole, x_t the slider's position with its corresponding velocity, and the last component encodes the angular velocity of the pole via the corresponding time derivatives of the angular observations. The state consists of

$$s_t = [x_t, \frac{\mathrm{d}}{\mathrm{d}t} x_t, \omega_t, \frac{\mathrm{d}}{\mathrm{d}t} \omega_t] \in \mathbb{R}^4,$$

i.e. includes not only the position and the angle, but also their corresponding derivatives. Thus, effective exploration of the state action space requires covering the position and velocity of the cart as well as the angles of the pole, and also entails the angular velocities of the pole, making the task significantly harder for constrained maximum entropy. An action $a_t \in \mathbb{R}$ is a real-valued action which is passed through a motor gear and then acts on the system.

Objective and Constraint: Definition & Implementation Details. After the state-action occupancy measure is estimated via the MLE in ($\hat{d}^\pi$-Est), we can estimate the entropy objective and the constraint.

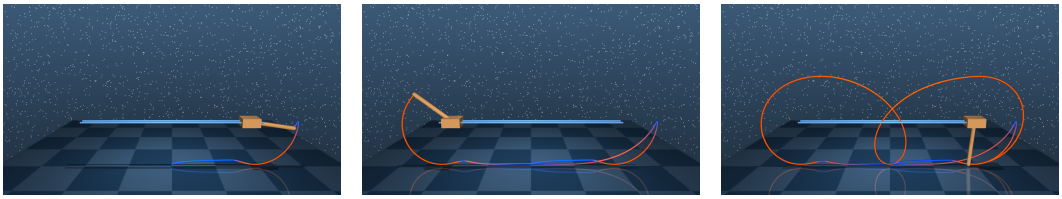
The constraint function is given by

$$c(s_t, a_t) \coloneqq \max(|x_t| - 2.0, 0),$$

where x_t encodes the slider's position at time $t \in [0, T_{\max}]$. For the sake of presentation, we also log the values $|x_t| - 2.0$ to make the constraint violation plot more expressive. The entropy and the cost function can be then computed by point-wise multiplications of the tensors given by: $\widehat{\lambda}$ evaluated on the state and action pairs, the logarithm of $\widehat{\lambda}$ for the objective and the constraint function values $(c(s_t, a_t))_t$ for each rollout, respectively. For numerical stability, we perform a log-sum-exp rescaling for these tensor computations. Afterwards we apply the squared penalty, sum both terms and obtain the pseudo-reward for which we compute the gradient using PyTorch's [102] automatic differentiation framework. The resulting gradient is plugged in the (PG-Thm) to compute the cumulative reward and optimize. We use a critic network as a baseline for the REINFORCE algorithm to improve stability, i.e. (PG-Thm) acts on the advantage function.

Detailed Analysis of the Results. We analyze separately:

- **Entropy and Constraint Violation.** In Figure 4a we observe a significant jump at $N = 1000$ in the entropy $\mathcal{H}$ which, combined with Figure 17 and additional logging information, corresponds to the policy understanding that symmetric discovery drastically improves the exploration in the state-action space. Afterwards, we observe a slight drop in the entropy again, as the policy needs to relearn, or more precisely combine, to turn the pole upwards and discover the space in a symmetric way. As evident in Figure 16 (bottom left), later episodes then learn how to push the penalty function right below the threshold to accelerate faster, i.e. get up the pole quicker and then discover even more. The final policy has almost perfect symmetry over the rollout horizon with varying angular velocities in the lower half of the angular part of the state space.
- **Controls.** Figure 16 shows that in particular early controls are slightly fuzzy and exhibit high(er) variance. In later iterations, i.e. $k \in [1000, 2500] \cap \mathbb{N}$, the policy $\pi^{(k)}$ correctly learns a bang-bang control style, which is optimal to turn the pole upward, especially since we do not penalize actions in our setting. This behavior is for larger k refined and the final policy perfectly flips the pole to the upper half of the angular's state space and is confirmed as the correct behavior by the reduced variance σ_θ^2, as it also improves on the tiny constraint violation at the end of the observation horizon.



(a) $t = 4.8$s (b) $t = 7.2$s (c) $t = 9.6$s

Figure 15: More fine-grained version of Figure 4b. Last iterate policy $\pi_{\theta^{(N)}}$ for **constrained maximum entropy exploration** on the SafeCartpole environment. The **tracer's color** encodes the (normalized) **pole's angular velocity** (dark blue for lower values and orange for higher values) which is part of the state space, i.e. has to be covered to maximize $\mathcal{H}$. A more detailed visualization over several training iterations is available in Figure 17. The corresponding **video** is available in the **supplementary material**.

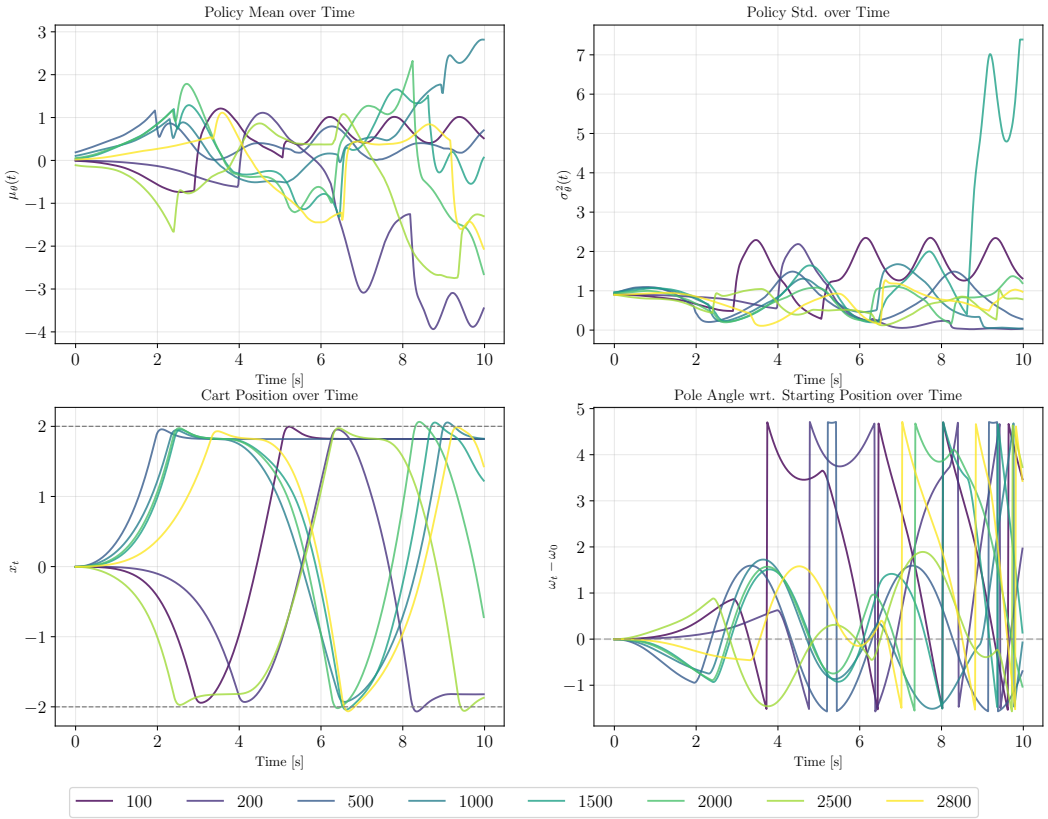


Figure 16: **Numbers** in the **legend** indicate the **iteration numbers** used in Figure 4a and Figure 17 and correspond to the same policy and the same seed.
Top: BoxTanhGaussian-Policy. We visualize the mean μ_θ (left) and the standard deviation σ_θ^2 (right) of the GMM policy π over the rollout horizon.
Bottom: Cart Position and Pole Angle. The left-hand side shows the cart's position over time with the dashed gray line indicating the threshold penalized by the constraint and the right-hand side shows the pole's angle relative to the angle of the starting position at $t = 0$. Discontinuities in the visualization arise since the values are invariant under modulo 2π.

Parameter	Value
Observation Space	$[-1.0, 1.0]^2$
Simulation Δ_t	0.02
Control Δ_t	0.02
Time limit	20s

(a) **PointMass environment.**

Parameter	Value
Pole Length	$[-0.1, 0.1]$
Gear	$[-1.0, 1.0]$
Slider Pos. Limit	$[-2.0, 2.0]$
Simulation Δ_t	0.01
Control Δ_t	0.01
Time limit	10s

(b) **SafeCartpole environment.**

Parameter	Value
Iterations / discount	3000 / $\gamma = 0.99$
Transitions / iteration	16384
B	8192
GMM update	200 steps; batch 512
Independent evaluation	Every 25 it.; 16+16 episodes
Gradient clipping	1.0

(c) **Training and evaluation.**

Model	Architecture
Policy π	$\mu_{\text{Net}}, \Sigma_{\text{Net}}$: [Lin(obs_dim, 256), Lin(256,256), Lin(256,act_dim)]
Occ. measure $\widehat{\lambda}$	GMM with $K = 16$, $(\sigma_k)_{k\in[K]} \subset [-5.0, 2.0]$
Critic / VF baseline	[Lin(obs_dim, 256), Lin(256,256), Lin(256,1)]

(d) **Model architectures.**

Table 4: Configurations for the continuous-control experiments in Section 6.2. Environment settings follow the RWRL benchmark [42]. The policy uses MLPs for the mean and diagonal covariance; VF denotes the critic baseline. PGP and PDPG use matched policy, GMM, transition, and seed budgets.

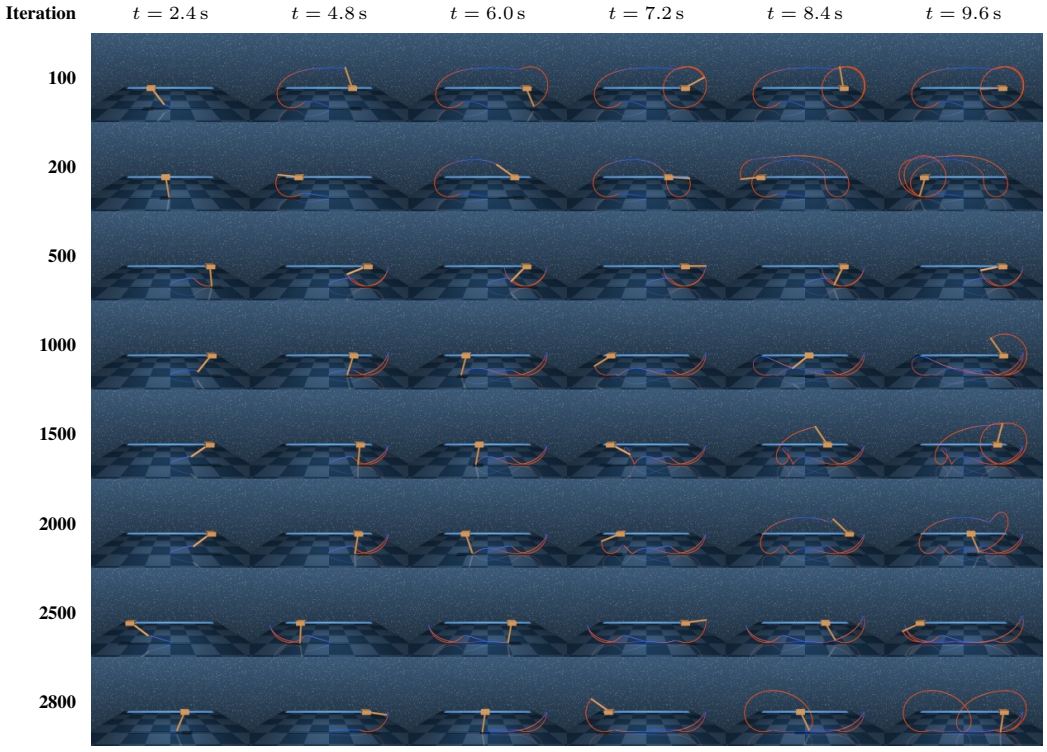


Figure 17: **Policy training evolution on SafeCartpole for a fixed seed.** Rows show training iterations and columns show trajectory times. The policy progressively learns to swing up the pole while respecting the safety constraint, cf. Figure 16. The tracer color encodes normalized pole angular velocity (lower to higher); corresponding videos are available in the supplementary material.

F Supplementary Material Details

We provide the full implementation of all algorithms and experiments presented in this paper. Components related to large-scale deployment, post-processing, and infrastructure-specific execution are omitted during review due to double-blind constraints. These components are auxiliary and do not affect the reproducibility of the proposed methods or the reported results. Experiments involving large-scale evaluation, ablation studies, and continuous control were executed on HPC resources. The underlying training code and all corresponding configuration files are provided and can be reproduced given suitable computational resources.

Computational Resources. We distinguish:

- **Grid World (Section 6.1).** Each run was executed on a single node with 20 CPU cores and 40 GB of memory, with an average wall-clock time of approximately 1h20min per configuration and seed. The full ablation comprises 600 configurations (penalty parameter, batch size, step size, environment variants and baseline searches), each evaluated over 10 random seeds, for a total of 6,000 runs. Runs were parallelized across 400 workers on the cluster, yielding a total wall-clock cost of roughly 20 hours and an aggregate CPU cost of approximately 1.6×10^5 core-hours.
- **Point Mass (Appendix E.3.2).** Each run was executed with 4 CPU cores and 8 GB of memory, with an average wall-clock time of approximately 30 minutes per configuration and seed. We evaluated 3 values of the constraint budget $\mathbb{R}_{\max}$ over 10 random seeds, for a total of 30 runs. The pretraining of the SAC reference policy π_{ref} was performed once and reused across all PointMass runs.
- **SafeCartpole (Appendix E.3.3).** Each run was executed on a single node with 20 CPU cores and 40 GB of memory, with an average wall-clock time of approximately 7 hours per configuration and seed. The ablation grid comprises 90 configurations evaluated over 10 random seeds, for a total of 900 runs and an aggregate CPU cost of approximately 1.3×10^5 core-hours.

The aggregate cost of the reported experiments is approximately 3×10^5 CPU core hours. Preliminary experiments and exploratory runs not reported in the paper.

F.1 Additional Material Folder: `/1_videos`

This folder contains rollout videos of the trained policy for $\text{seed} = 2$, corresponding to Figure 17. Each video follows the **naming convention**

- `seed2_iter<k>.mp4`,

where `<k>` denotes the training iteration. We note a minor discrepancy between the time axis in Figure 17 and the video durations. This is due to the frames-per-second rendering configuration used in the visualization software of Dulac-Arnold et al. [42] and does not affect the underlying policy behavior.

F.2 Source Code Folder: `/2_code`

The source code is organized into discrete and continuous control experiments. Installation instructions for both settings are provided in their individual `INSTALLATION.md`-file.

F.2.1 Grid World Experiment: `/1_gridWorld`

The `<config>.json` files specify the experimental configurations for all figures in the main paper as well as the ablation studies reported in the appendix. Each configuration file has a corresponding `<config>_run.py` script that executes the experiment; all runs can also be invoked via the provided `Makefile`.

The **utility files** are organized as follows:

- `agent.py` implements the optimization methods proposed and evaluated in the paper.
- `experiments.py` provides a wrapper around the Gym interface [21], handling checkpoints, logging, and post-training evaluation.

- `frozenlake_utility.py` contains helper functions for policy debugging and visualization.
- `general_utility.py` implements objective functions (e.g. entropy $\mathcal{H}$), count-based occupancy-measure estimation, importance sampling weights, and gradient computations.

F.2.2 PointMass & SafeCartpole Experiment: `/2_continuous`

The following **utility files** are provided in the folder:

- `pgp_core.py` contains the model architectures, rollouts storage data structures and wrapper for environment interactions.
- `point_mass_main.py` Train the SAC reference policy and train the PGP policies given the KL divergence constraint.
- `main.py` corresponds to the SafeCartpole environment and defines the training loop.
- `debug_utils.py` renders and visualizes intermediate policies during training.
- `metrics_utils.py` logs training metrics (e.g. entropy, constraint violation) to the console and HPC log files.
- `pendulum.py` implements a wrapper around the MuJoCo model for the SafeCartpole environment.
- `policy_utils.py` provides helper functions for saving and loading training checkpoints.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction accurately reflect the paper's contributions. The three contributions (C1)-(C3) listed in the introduction correspond directly to the algorithm in Section 5 (PGP), the convergence analysis in Section 5.1 Section 5.2 and the empirical validation in Section 6

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Limitations are discussed in Section 7 and in the discussion following Corollary 5.6. Specifically, we discuss the relatively large $\widetilde{\mathcal{O}}\left(\varepsilon^{-6}\right)$ sample complexity, stemming from the $\mathcal{O}\left(\varepsilon^{-4}\right)$ batch size requirement, and the gap between theory and practice in step-size and batch-size selection.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: All theoretical results are accompanied by complete proofs. Assumption 5.1, Assumption 5.2, Assumption 5.3, and Assumption 5.5, are clearly stated in Section 5.1, and all theorems and lemmas reference the relevant assumptions. Full proofs are provided in Appendix B (convergence analysis), Appendix C (density result and anchored softmax parametrization), and Appendix D (unified penalty framework). Proof sketches are provided in the main text where the full proofs are deferred.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: All experimental details are disclosed in Section 6 with all details available in Appendix E. The grid-world setup, hyperparameter ranges, and ablation protocols are provided in Appendix E.2. The continuous-control settings, including environment parameters, model architectures, and training hyperparameters, are tabulated in Table 4 of Appendix E.3. Source code is provided in the supplementary material with detailed information available in Appendix F.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: The full source code for all experiments is provided in the supplementary material Appendix F, with installation instructions in `INSTALLATION.md`. The grid-world experiments can be reproduced via the provided `makefile` and corresponding configuration files. The continuous-control experiments are reproducible via `make experiment`. No external datasets are used; all environments are standard benchmarks (FrozenLake from [21], PointMass and SafeCartpole from the [42]).

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

Answer: [Yes]

Justification: Training and evaluation details, including data collection (trajectory rollouts), hyperparameters (penalty parameter, batch size, step size, trajectory length), optimizer choices, and model architectures, are fully specified in Appendix E, with consolidated configurations in Table 4.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: All entropy and constraint-violation results are reported as $\mu \pm \sigma$ over 10 random seeds, as stated at the beginning of Section 6 and consistently in all figure captions (e.g. Figure 1, 2, 4a). Shaded bands in plots represent one standard deviation.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Compute resources, including CPU core counts, memory, per-run wall-clock times, total run counts, parallelization strategy, and aggregate core-hours, are reported in Appendix F. We additionally disclose that preliminary and unreported runs account for an additional compute cost of comparable order of magnitude.

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research conforms with the NeurIPS Code of Ethics. The work is theoretical and methodological in nature, with experiments conducted on standard simulation benchmarks, and does not involve human subjects, sensitive data, or applications with foreseeable harm.

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: This work is a foundational contribution to the theory and methodology of constrained reinforcement learning, focused on convergence guarantees for policy-gradient methods under occupancy-measure constraints. It is not tied to any specific deployment or application, and we do not foresee a direct path from our methodology to negative societal impacts beyond those generally associated with reinforcement learning research.

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper does not release pre-trained models, datasets, or other assets that pose a risk of misuse. The released code is a reference implementation of an optimization algorithm evaluated on standard simulation environments.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

Justification: We use [21] (MIT license), [117] (Apache 2.0 license), [114] (Apache 2.0 license), [105] (MIT license) [102] (BSD-style license) and [42] (Apache 2.0 license).

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The released source code (Appendix F) is documented with installation instructions, configuration files for reproducing each figure, and inline documentation of the main components. The code is anonymized for double-blind submission and will be released under CC BY 4.0 license.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing or research with human subjects.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve research with human subjects.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: LLMs were not used as a component of the core methodology, scientific contribution, or experimental design of this work.