SACrificing Intuition: Kullback-Leibler Regularized Actor-Critic

Anonymous authors
Paper under double-blind review

Abstract

One of the most popular algorithms in reinforcement learning is Soft Actor-Critc (SAC), as it promises to elegantly incorporate exploration into the optimization process. We revisit SAC through the lens of constrained optimization and develop Kullback-Leibler Actor-Critic (KLAC), a principled extension of Soft Actor Critic that replaces the heuristic entropy bonus of SAC with a Kullback-Leibler regulariser against an arbitrary reference policy. We contrast Kullback-Leibler Actor Critic with Soft Actor Critic and demonstrate analytically and with a concrete counterexample that injecting the entropy term directly into the reward, as implemented in Soft Actor Critic, violates the convexity assumptions of the dual proof of near-optimality and can render the learned policy arbitrarily sub-optimal no matter how small the temperature is chosen. This understanding reveals a fundamental systemic flaw in SAC, especially for sparse reward environments. To retain the empirical exploration benefits without sacrificing theoretical soundness, we introduce a fixed uniform reward bias that captures the intrinsic motivation effect to *stay alive*. Additionally, we propose a Kullback-Leibler annealing schedule that unifies discrete and continuous action spaces by mapping an intuitive probability of exploitation to a closed-form entropy or Kullback-Leibler target. Together, these contributions yield an algorithm that at least matches the sample efficiency and performance of Soft Actor Critic as demonstrated on MuJoCo and MinAtar benchmarks while enjoying provable near optimality, interpretable hyperparameters, and a theoretically grounded exploration mechanism. We provide code to reproduce all plots in the paper.

1 Introduction

Deep reinforcement learning (RL) has achieved remarkable empirical success in domains ranging from games to continuous control, yet our theoretical understanding of when these methods work (or fail) often lags behind practice (Barto, 2021). In classical tabular RL settings, strong optimality guarantees could be obtained under convexity assumptions, but modern deep RL algorithms with nonlinear function approximators are much harder to analyze rigorously. As a result, many state-of-the-art algorithms prioritize empirical convergence over optimality, and identifying systematic weaknesses in their formulations requires renewed theoretical scrutiny. One important area of inquiry is regularized or entropy-maximizing RL, which augments the standard objective with bonus rewards to encourage exploration and improve stability. A prominent example is Soft Actor-Critic (SAC) (Haarnoja et al., 2018), an off-policy actor-critic algorithm that maximizes a weighted sum of environment reward and policy entropy. SAC has demonstrated state-of-the-art performance on a range of continuous control benchmarks while maintaining high training stability across random seeds. Understanding precisely why SAC is stable, how its entropy injection in the actor and critic shapes exploration, and what optimality cost this regularisation imposes is therefore critical both for safe deployment and for principled algorithmic improvements. By aiming to succeed at the task while acting as randomly as possible, SAC embodies the maximum entropy RL framework and is often viewed as a principled approach to balance exploration and exploitation (cf. Levine, 2018).

The theoretical justification for SAC's entropy augmentation beyond the intuitive explanation (Arriojas et al., 2023) continues to be challenged in the field. In contrast to methods that impose an explicit divergence constraint on policy updates (Peters et al., 2010; Schulman et al., 2015; Geist et al., 2019), SAC adds the

entropy regularizer directly to the reward function. We revisit SAC through the lens of constrained policy optimization and find that this design choice has significant consequences. In fact, injecting the entropy term into the reward fundamentally alters the structure of the optimization problem, breaking the separability and convexity properties that underpin theoretical guarantees in regularized RL (Pacchiano et al., 2021b). We show analytically that SAC's objective is no longer equivalent to a true KL-regularized policy search. Consequently, the usual near-optimality guarantees no longer hold and SAC can converge to an arbitrarily suboptimal policy. We derive a concrete counterexample illustrating SAC's failure mode of softening the rewards with entropy, where the agent can become biased towards high-entropy behavior that yields zero real reward, even when a far better policy exists. In essence, incorporating entropy into the reward can dominate the learning signal and prevent convergence to the optimal policy, a systemic flaw in SAC that is especially severe in sparse-reward or long-horizon tasks. Empirical evidence has hinted at this issue for some time already. For example, SAC is known to become unstable without additional constraints in some settings as in discrete-action domains, where SAC requires ad-hoc modifications for stable learning (Xu et al., 2021b;a). Other recent studies have observed that removing or reducing the entropy bonus can actually improve performance in practice (Yu et al., 2022a). These observations, which so far lack a theoretical explanation, are supported by our analysis of how entropy shaping can misalign the objective from the true task reward.

We introduce the Kullback-Leibler Actor-Critic (KLAC) algorithm, a principled variant of SAC that resolves these theoretical and practical issues. KLAC is derived from the constrained optimization view of RL and imposes an explicit KL regularization against a reference policy (e.g., the uniform distribution) at each update. This approach can be seen as a direct application of relative entropy policy search (Peters et al., 2010) in an actor-critic setting, ensuring that policy updates are conservative and grounded in convex duality. Critically, KLAC's objective leads to the same kind of softmax policy update as SAC, but without corrupting the reward signal. We prove that the optimal policy under our KL regularizer remains near-optimal with respect to the true (unregularized) return, recovering the performance bound characteristic of sound regularized RL methods (Pacchiano et al., 2021b).
In summary, the main contributions of this paper are:

1. A critical theoretical analysis of the SAC algorithm, proving that its built-in entropy reward can lead to non-convex optimization and arbitrarily suboptimal policies.

2. KLAC, a novel actor-critic algorithm that replaces SAC's heuristic entropy bonus with a principled KL divergence regularization. KLAC is derived from first principles and comes with a guarantee of near-optimality.

3. An empirical evaluation of KLAC against SAC on standard continuous control benchmarks (MuJoCo) and a suite of discrete-action tasks (MinAtar).

We hope that our work sheds light on the importance of properly formulating regularization in deep RL and offers a practical solution that is both theoretically grounded and effective in practice. We provide open-source code for KLAC to facilitate reproducibility and further research. The rest of the article is structured as follows: Section 2 introduces the necessary mathematical background to our work. Section 3 presents related work on the theoretical and practical side. Section 4 presents a theoretical analysis of SAC. Section 5 presents theoretical advancements and the new KLAC algorithm that we derive. Section 6 introduces the experiments used and discusses their results. Section 7 presents future work and summarizes our article.

2 Background: Regularized Policy Search

In this section, we provide the necessary background to the reinforcement learning problem and the corresponding constrained optimization formulation.

2.1 Markov Decision Process

We model the agent–environment interaction as a discounted Markov decision process (MDP) $M = \langle S, A, P, R, \gamma \rangle$(Bellman, 1957). The transition kernel $P : S \times A \times S \to [0, 1]$ satisfies

$$\sum_{s' \in S} P\big(s' \mid s, a\big) \;=\; 1 \qquad \forall\, (s, a) \in S \times A$$

with the state space set S and the action space A. The one-step reward function is $R : S \times A \to \mathbb{R}$ simplified as $r(s, a)$,
and $0 \leq \gamma < 1$ is the discount factor. A (stochastic) policy is a conditional distribution $\pi : S \times A \to [0, 1]$ obeying

$$\sum_{a \in A} \pi\big(a \mid s\big) \;=\; 1 \qquad \forall\, s \in S\,.$$

Given an initial state s_0 , the discounted return under π is

$$G_{s_0} \;=\; \sum_{t=0}^{\infty} \gamma^t\, r(s_t, a_t)\,, \qquad a_t \sim \pi(\,\cdot \mid s_t),\; s_{t+1} \sim P(\,\cdot \mid s_t, a_t)\,.$$

The control objective is to find an optimal policy

$$\pi^\star \;\in\; \arg\max_{\pi}\, \mathbb{E}\big[G_s\big] \qquad \text{for every } s \in S\,,$$

where the expectation is over trajectories induced by π. This formulation underpins the regularised policy-search framework introduced next.

2.2 Regularised Policy Search and Convex Duality

We begin by framing the infinite–horizon, γ–discounted control problem as a linear program (LP) over occupancy measures. Let $\mu_0 : S \to [0, 1]$ denote an initial state distribution with $\sum_s \mu_0(s) = 1$. For any stationary policy π, its γ-discounted state–action occupancy measure is

$$\mu_\pi(s, a) := (1 - \gamma) \sum_{t=0}^{\infty} \gamma^t \Pr_{\pi}\big[S_t = s,\, A_t = a \mid S_0 \sim \mu_0\big], \qquad (s, a) \in S \times A\,.$$

Intuitively, $\mu_\pi(s, a)$ is the (discounted) fraction of time the process spends in (s, a). The flow-conservation constraint below formalises a discounted balance law. The probability mass arriving at each state equals the initial inflow plus discounted transitions from everywhere else. The normalisation $\sum_{s,a} \mu_\pi(s, a) = 1$ follows from the $(1 - \gamma)$ factor and makes the feasible set compact. Any feasible μ induces a unique stationary policy via $\pi(a \,|\, s) = \mu(s, a) / \sum_{a'} \mu(s, a')$ whenever the denominator is nonzero. Following Belousov & Peters (2017); Nachum et al. (2018); Pacchiano et al. (2021a), we introduce a convex regulariser in the objective with an explicit weight $\alpha > 0$, which enriches the linear program (LP) with a convex regulariser that captures additional design preferences such as trust-region proximity, entropy-driven exploration, sparsity, or risk sensitivity. With μ_π the infinite-horizon control problem is

$$\begin{aligned} &\max_{\mu_\pi}\; J_\alpha(\mu_\pi) = \sum_{s \in S} \sum_{a \in A} \mu_\pi(s, a)\, r(s, a) - \alpha\, F(\mu_\pi) \\ \text{s.t.}\quad &\sum_{a' \in A} \mu_\pi(s', a') = (1 - \gamma)\mu_0(s') + \gamma \sum_{s \in S} \sum_{a \in A} \mu_\pi(s, a) P(s' \mid s, a)\;,\; \forall s' \in S, \\ &\sum_{s,a} \mu_\pi(s, a) = 1, \quad \mu_\pi(s, a) \geq 0 \;\; \forall s, a\,. \end{aligned} \tag{1}$$

Here $\mu_\pi(s, a)$ captures the discounted visitation frequency of (s, a). The first (flow-conservation) constraint balances discounted in-flow and out-flow for every state, while the normalisation $\sum_{s,a} \mu_\pi(s, a) = 1$ keeps the

feasible set compact and guarantees a one-to-one correspondence between any feasible μ_π and its induced policy π. The regularizer $F : \mathcal{D} \subseteq \mathbb{R}^{|S||A|} \to \mathbb{R} \cup \{\infty\}$ is proper, convex, and lower semicontinuous on a convex domain $\mathcal{D}$ that contains the feasible occupancies. The weight α tunes the reward–regularity trade-off in the terminology of inverse problems (Tikhonov & Arsenin, 1977). Using the Fenchel conjugate $F^*(u) := \sup_{x\in\mathcal{D}}\{\langle x, u\rangle - F(x)\}$ we can apply the Fenchel–Rockafellar duality according to Eq.(1) and obtain the dual

$$\min_{v\in\mathbb{R}^{|S|}} J_D(v) \;=\; (1-\gamma)\sum_{s\in S} v_s\,\mu_0(s) \;+\; F^*\big(\mathbf{A}^\top \boldsymbol{v}\big)\;, \tag{2}$$

where the advantage vector $\mathbf{A}^\top \boldsymbol{v} \in \mathbb{R}^{|S|\times|A|}$ has components

$$\big(\mathbf{A}^\top \boldsymbol{v}\big)_{s,a} = r(s,a) + \gamma \sum_{s'} P(s' \mid s,a)\, v_{s'} - v_s\;.$$

The second term $F^*\big(\mathbf{A}^\top \boldsymbol{v}\big)$ in the dual objective Eq.(2) is the penalisation that enforces the structural preferences encoded by the primal regulariser F. For example, choosing F as a KL divergence makes F^* the log-partition function and recovers the soft-max (entropy-regularised) policy update. The vector $\mathbf{A}^\top \boldsymbol{v}$ itself is the one-step temporal-difference (TD) error. It is positive when action a is better than the current value estimate v_s and negative otherwise. Minimising the dual therefore pushes all advantages into the domain where F^* is finite, yielding the unique value vector $v^\star$ and, via $\mu^\star = \nabla F^*(\mathbf{A}^\top \boldsymbol{v}^\star)$, the policy that maximises reward while respecting the chosen regularisation. An equivalent way to encode preferences is the constrained form

$$\max_{\mu} \sum_{s,a} \mu(s,a)\, r(s,a) \quad \text{s.t.} \quad \text{flow/normalisation as above,} \qquad F(\mu) \le \tau\;, \tag{3}$$

with budget $\tau > 0$, which is known as Ivanov regularisation (Ivanov, 1962). The Lagrangian introduces a multiplier $\lambda \ge 0$ for $F(\mu) \le \tau$ and yields the partial dual

$$\min_{v,\;\lambda\ge 0} (1-\gamma)\langle v, \mu_0\rangle \;+\; \lambda\,\tau \;+\; \lambda\, F^*\Big(\tfrac{1}{\lambda}\mathbf{A}^\top v\Big)\;.$$

Eliminating λ recovers the penalised dual Eq.(2) with $\alpha = \lambda$. Under the conditions that F is strictly convex and coercive on the feasible set, the mapping $\alpha \mapsto F(\mu_\alpha)$ is continuous and strictly decreasing, so for any target τ there exists a unique α such that the penalised solution satisfies $F(\mu_\alpha) = \tau$. In practice one can choose τ and perform binary search over α until $F(\mu_\alpha) \approx \tau$. This is the standard Tikhonov-Ivanov equivalence exploited widely in ML, such as support vector machines (Oneto et al., 2016).

2.3 Soft Actor-Critic

We briefly recapitulate the Soft Actor-Critic (SAC) algorithm, an off-policy actor–critic method that augments the cumulative discounted reward with an entropy incentive to favour stochastic behaviours (Haarnoja et al., 2018). At every decision point the agent therefore seeks to maximise the expected return and the randomness of its action choices, leading to policies that explore proactively without abandoning high-value regions of the state space.

SAC employs a pair of parameter-tied critics $Q_{\theta_1}, Q_{\theta_2}$ updated toward bootstrapped targets that include the current policy's entropy term; taking the minimum of the two critics mitigates over-estimation bias. The actor π_ϕ is obtained by minimising the Kullback–Leibler divergence between $\pi_\phi(\cdot \mid s)$ and the Boltzmann distribution induced by the critic ensemble, yielding a closed-form stochastic gradient. A novel automatic temperature adjustment tunes the entropy coefficient α so that the realised entropy tracks a user-specified target throughout training (Haarnoja et al., 2018). In practice, these ingredients have three advantages: (i) high sample efficiency from off-policy replay, (ii) robustness across reward scales through adaptive α, and (iii) numerical stability thanks to the twin-critic safeguard.

Despite these strengths, SAC departs from classical regularised policy search in one crucial aspect: it injects the entropy bonus directly into the per-step reward processed by the critic rather than enforcing a divergence

constraint solely in the actor. As we demonstrate in Section 4, this design choice breaks the separability and convexity properties required for standard duality-based optimality guarantees and can bias learning toward high-entropy yet low-reward solutions. The next section situates this observation within the broader literature on regularised control.

3 Related Work

This related work section is divided into three parts. The first part presents theory behind regularized policy search. The second part presents practical algorithms that are in the realm of regularized policy search algorithms. The third part is examining papers that identified shortcomings of SAC and where it fails.

3.1 Theory behind Regularized Policy Search

MDP control can be cast as a linear program over state–action occupancy measures, whose dual variables yield the value function and guarantee strong duality under mild conditions (Manne, 1960). Incorporating expectation constraints produces the standard constrained-MDP LP (Altman, 2021; Paternain et al., 2019). Safe-RL algorithms such as Constrained Policy Optimization exploit this primal-dual structure, using KL-regularized trust-region updates and multiplier tuning to achieve (approximate) constraint satisfaction and convergence (Achiam et al., 2017). In the policy search literature, Peters et al. (2010) introduced Relative Entropy Policy Search (REPS), which imposes a KL-constraint between the updated policy and a reference policy at each iteration. By solving a convex dual problem, REPS guarantees small policy updates and yields an analytic policy improvement step. Natural Actor-Critic (NAC) methods Peters & Schaal (2008) also fall into this line of work, as they leverage natural gradients under a KL geometry to improve stability and efficiency of policy updates. Notably, Trust Region Policy Optimization (TRPO) (Schulman et al., 2015) and its successors like PPO (Schulman et al., 2017) enforce a soft KL limit (or introduce gradient clipping) on policy changes to ensure monotonic improvement and prevent divergence. These KL-regularized or entropy-regularized approaches were later grounded in theory connecting them to mirror descent and convex optimization (Geist et al., 2019; Neu et al., 2017). (Geist et al., 2019) introduce a regularized Bellman operator for an arbitrary convex regularizer (e.g., negative entropy or an f-divergence) and show that the optimal policy can be characterized by a Fenchel–Legendre transform of the value function. Building on this, (Vieillard et al., 2020) analyze the specific case of KL regularization in approximate dynamic programming. They prove that incorporating a relative-entropy penalty implicitly performs a form of Q-value averaging, which can reduce overestimation and stabilize training. Crucially, convex duality provides the mathematical bridge between adding a regularizer in the primal LP (occupancy measure) formulation and the emergence of softmax policies or advantage functions in the dual formulation (Belousov & Peters, 2017). A paradigmatic example is given by (Pacchiano et al., 2021a), who study the theoretical properties of policy optimization under a KL regularization penalty. Similarly, (Zahavy et al., 2021) consider a broad class of convex MDP objectives, where the goal is to optimize a convex function of the stationary distribution (occupancy measure) rather than a linear reward by recasting the problem as a zero sum two player game and derive a meta-algorithm connecting many fields, such as apprenticeship learning, pure exploration and constrained RL. Our work shows that SAC is actually connected to these works, but breaks a fundamental property of those. We show theoretically and with an example why this error may be catastrophic (and why it is often times not) and recover a more general algorithm that performs better and is more intuitive to use. Further our theoretical findings support evidence that other papers already gathered but could not explain on a theoretical level.

3.2 Algorithms using Regularized Policy Search

Overall, KL-regularized policy search has become one of the fundamental algorithms in RL, underpinning a spectrum of on-policy, off-policy, imitation, offline, and hierarchical methods, and inspiring alternative divergence-based approaches that continue to advance the field.

KL divergence has been widely used to stabilize policy updates in reinforcement learning by constraining the change between successive policies. Early work introduced Relative Entropy Policy Search (REPS),

which enforces a hard bound on the KL divergence between the new and current policy to guarantee stable and monotonic improvement (Peters et al., 2010; Kakade & Langford, 2002). REPS derives a closed-form exponential-weight update from the convex dual, so the policy step size is controlled by a single Lagrange multiplier instead of an external learning-rate schedule. Building on this principle, Trust Region Policy Optimization (TRPO) by Schulman et al. (2015) further enforces a KL constraint on the policy update step, leading to improved empirical stability in deep RL benchmarks. TRPO formulates the update as a constrained quadratic program and solves the KL trust-region with conjugate-gradient plus line-search, ensuring first-order monotone improvement. Proximal Policy Optimization (PPO) refines this idea by replacing the hard constraint with either a clipped surrogate objective or an adaptive KL penalty, simplifying implementation while retaining performance (Schulman et al., 2017).

KL regularization has also been incorporated into actor–critic and off-policy methods. ACER couples truncated importance sampling with a KL correction term, giving an unbiased yet variance-reduced off-policy gradient that is numerically stable under replay (Wang et al., 2017). ACKTR approximates the natural gradient using Kronecker-factored blocks of the Fisher matrix, yielding curvature-aware parameter updates without forming the full Hessian (Wu et al., 2017). Maximum a Posteriori Policy Optimization (MPO) casts the update as an EM algorithm optimizing a KL-regularized objective, fitting a parametric policy to an advantage-weighted action distribution (Abdolmaleki et al., 2018). Soft Actor–Critic (SAC) maximizes a combination of expected return and policy entropy, which can be seen as KL regularization to a uniform prior and uses the re-parameterisation trick to differentiate through stochastic actions (Haarnoja et al., 2018).

In imitation learning and offline RL, KL-based regularization is crucial for staying close to reference behaviors (Ho & Ermon, 2016; Vinyals et al., 2019; Wu et al., 2019; Kumar et al., 2019; Ashvin et al., 2020).

Hierarchical RL methods such as HiREPS extend REPS to a two-level policy hierarchy by applying KL constraints at both the skill and meta-policy levels, enabling stable learning of diverse sub-policies Daniel et al. (2016). More recently, optimal transport trust region policy optimization replaces the KL constraint with a Wasserstein-distance trust region, addressing limitations of KL when policy supports have little overlap Terpin et al. (2022). Finally, generalizations to other f-divergences and entropy regularization have been explored to balance exploration and exploitation in policy search Williams & Peng (1991); Levine (2018).

3.3 Previous Critiques of Soft Actor Critic

Although entropy regularization is intended to improve learning stability, it can sometimes introduce new instability if not carefully managed. SAC adjusts a temperature parameter α to target a desired entropy level (Haarnoja et al., 2018). If this target entropy is set improperly or the reward signal is sparse, the automatic temperature adjustment can oscillate or diverge. An excessively large entropy bonus (high α) drives the policy to act almost randomly, making the value estimates difficult to stabilize. On the other hand, a very low α nullifies the exploratory benefit as the policy is not induced with further entropy regularization. (Wang & Ni, 2020) show that a poorly tuned entropy target can break the value-entropy trade-off, causing divergence. In discrete-action domains, (Xu et al., 2021a) observed SAC training to be highly unstable with rapid shifts in the policy's entropy when no constraint is placed on policy updates. They conjecture that the lack of a trust-region or KL constraint in SAC allows the policy to change too abruptly, causing the critic's target values to keep moving (since the target includes an entropy term that changes with the policy). This leads to oscillation and learning instability. Indeed, recent variants for discrete SAC add constraints or schedule the entropy coefficient to mitigate these issues (Wang & Ni, 2020; Xu et al., 2021a; Wei et al., 2025). Furthermore, SAC's use of "double Q-learning" (two Q-networks with minima) to control overestimation can introduce biases in value estimation as some studies report systematic underestimation that slows learning (Ciosek & Whiteson, 2019; Pan et al., 2020). Other work has directly questioned the necessity of the entropy reward bonus and found that removing it helps, but without providing an explanation (Yu et al., 2022b). Without additional regularization, the entropy bonus or choice of Q-network can cause SAC to underperform or even diverge in complex environments with deceptive rewards or very sparse rewards where uncontrolled entropy maximization leads to aimless exploration.
This article extends these existing findings by identifying the entropy bonus term in the reward formulation as the theoretical and practical problem, which helps explaining these observations. By explicitly isolating

the entropy term, we highlight a key mechanism through which SAC's stability can break down, offering a clearer direction for designing alternative regularization strategies.

4 Theoretical Analysis of SAC

Here, we provide a theoretical analysis of SAC under the linear program regime and formulate the maximum entropy objective as a constrained optimization problem. We then go into detail on how the practical implementation of applying an entropy reward bonus to the reward function itself breaks optimality on a theoretical level and show how it breaks in practice as well with an illustrative example.

4.1 SAC as Linear Program

We next reinterpret Soft Actor–Critic (SAC) within the linear–programming framework. The crucial distinction is that SAC adds the entropy bonus to the Bellman target, thereby coupling the temperature parameter α with both the critic and the actor updates. Concretely, the SAC critic is trained towards

$$\hat{Q}_{\mathrm{SAC}}(s,a) = r + \gamma\, \mathbb{E}_{a' \sim \pi_\alpha(\,\cdot\,|s')}\big[Q_{\bar{\theta}}(s',a') - \alpha \log \pi_\alpha(a' \mid s')\big]\,,$$

which already contains the Lagrange multiplier α that also appears later in the actor projection of the critic. To illuminate the underlying optimisation, recall the Lagrangian employed by SAC when the entropy is treated as the constraint

$$\begin{aligned}\mathcal{L} = &\sum_{s,a} \mu_\pi(s)\,\pi(a \mid s)\, R_s^a + \alpha\Big(-\sum_{s,a} \mu_\pi(s)\,\pi(a \mid s) \log \pi(a \mid s) - H_{\min}\Big) \\ &- \sum_{s'} \boldsymbol{\theta}^\top \boldsymbol{\varphi}_{s'} \Big(\mu_\pi(s') - \sum_{s,a} \mu_\pi(s)\,\pi(a \mid s) P_{ss'}^a\Big) - \lambda\Big(1 - \sum_{s,a} \mu_\pi(s)\,\pi(a \mid s)\Big)\,,\end{aligned}$$

where the Lagrangian multipliers for the state flows are denoted by the parameter vector $\boldsymbol{\theta}$ of our value function, and φ_s is the state representation or feature function. Differentiating w.r.t. the discounted occupancy measure yields

$$\frac{\partial \mathcal{L}}{\partial d_\pi(s,a)} = R^{sa} - \alpha\big(\log \pi(a \mid s) + 1\big) + V(s) - \gamma\, \mathbb{E}_{s' \sim P^{s,a}}\big[V(s')\big] - \beta\,.$$

In a strict actor–critic decomposition, the $-\alpha \log \pi$ term belongs exclusively to the actor as it encodes the relative-entropy constraint that moderates policy updates (Peters et al., 2010; Pacchiano et al., 2021a). SAC, however, introduces the same term again in the critic target, coining the term soft-value and using a soft-Bellman operator, penalizing the critic in addition to the actor regularization through the constraint. However, duplicating the entropy penalty in both the actor and critic overemphasises the constraint, provides no additional regularisation benefit, and increases the duality gap relative to the primal optimum. The redundancy is amplifying the entropy signal inside the value estimate, it can bias policy evaluation and ultimately hinder convergence in sparse-reward settings.

4.2 Loss of Convexity and Duality Gap

We now demonstrate that adding the entropy bonus to the reward breaks the convex structure required for the standard KL-regularised analysis and can drive SAC arbitrarily far away from optimality. Let the shaped reward be $\tilde{r}(s,a) = r(s,a) - \alpha \log \pi(a \mid s)$ with temperature $\alpha > 0$, and $\pi_\alpha^{\mathrm{SAC}}$ denotes any fixed-point policy of SAC whose critic target is based on $\tilde{r}$. We claim that

$$\exists\ \text{MDP}\ M,\ \forall \alpha > 0: \qquad J_M\big(\pi_\alpha^{\mathrm{SAC}}\big) < J_M^\star - \varepsilon\,,$$

where ε equals the reward dynamic range $[\max r - \min r]$, and J_M denotes the expected performance of a policy in MDP M. Thus, unless $\alpha = 0$, the entropy term can leave the learned policy arbitrarily sub-optimal.

To highlight the source of the sub-optimality we fix a stationary policy π and write its γ-discounted occupancy measure as $\mu_{s,a} = (1-\gamma)\sum_{t \geq 0} \gamma^t \Pr_\pi[s_t = s, a_t = a]$ with $\mu_s = \sum_a \mu_{s,a}$. Because $\pi(a \mid s) = \mu_{s,a}\mu_s$, the

State	Action	Next state / reward
s_0	a_0	$s_T,\ R > 0$
s_0	$a_i\ (1 \le i \le N)$	$s_0,\ 0$
s_T	–	$s_T,\ 0$

Table 1: A two-state MDP with one action to transition to the terminal state and $N-1$ actions to stay in the same state, avoiding termination of the MDP.

cumulative entropy bonus along a trajectory equals $\sum_{s,a} \mu_{s,a}\big[-\log \pi(a \mid s)\big]$. Maximising the shaped return therefore becomes

$$\max_{\mu \ge 0} \sum_{s,a} \tilde{r}_{s,a}\, \mu_{s,a} - \alpha \sum_{s,a} \mu_{s,a} \log \frac{\mu_{s,a}}{\mu_s} \ . \tag{4}$$

By contrast, the primal objective for KL-regularised RL is

$$\max_{\mu \ge 0} \sum_{s,a} r_{s,a}\, \mu_{s,a} - \alpha \sum_{s,a} \mu_{s,a} \log \frac{\mu_{s,a}}{q_{s,a}} \ ,$$

where the reference measure $q_{s,a}$ renders the regulariser separable across state–action pairs. In that convex setting the duality proof yields $J^\star - J(\pi_\alpha) \le \alpha(1-\gamma)^{-1} \log |A|$ (Pacchiano et al., 2021a).

The denominator μ_s in Eq.(4) couples all actions within each state, giving

$$-\alpha \sum_{s,a} \mu_{s,a} \log \mu_{s,a} + \alpha \sum_{s} \mu_s \log \mu_s \ .$$

Since the mixed second-order derivatives $\partial^2\big[\mu_s \log \mu_s\big]/(\partial \mu_{s,a} \partial \mu_{s,b}) = \alpha/\mu_s$ for $a \neq b$ are non-zero, the objective is neither linear nor separable convex in μ. Consequently, the convex-duality gap bound that underpins near-optimality for KL-regularised methods no longer applies for SAC. Inserting the entropy term into the reward destroys both separability and convexity, explaining the potentially unbounded sub-optimality of SAC and motivating the corrections developed.

4.3 Practical Implications

We now show an example, where a soft optimal policy may never find the true optimum of the original MDP, validating our claim and theoretic findings. Consider a finite MDP as the one in Table 1, with discount $0 < \gamma < 1$ and two states. Start by introducing two possible polices.

$$\pi_{\text{exit}}(a_\star \mid s_0) = 1, \quad \pi_{\text{loop}}(a_i \mid s_0) = \tfrac{1}{N}\ (1 \le i \le N) \ .$$

We now present an example in which a soft optimal policy fails to recover the true optimum of the original MDP, thereby supporting our claim and theoretical results. We consider the finite MDP in Table 1, with discount factor $0 < \gamma < 1$ and two states. We introduce two candidate policies

$$\pi_{\text{exit}}(a_\star \mid s_0) = 1, \quad \text{and} \quad \pi_{\text{loop}}(a_i \mid s_0) = \tfrac{1}{N}\ (1 \le i \le N) \ .$$

In the primal problem we can easily deduce that the true returns of each policy are as follows $J(\pi_{\text{exit}}) = R,\ J(\pi_{\text{loop}}) = 0$. The policy π_{exit} immediately goes into the rewarding state and is optimal, whereas the looping policy never receives reward. If we introduce the shaped reward function with entropy bonus we observe the contrary. While the agent remains in s_0 under π_{loop}, it collects per-step entropy $\log N$ as reward. We observe that the collected reward now becomes

$$J_\alpha(\pi_{\text{exit}}) = R + \alpha \log N, \quad \text{and} \quad J_\alpha(\pi_{\text{loop}}) = \frac{\alpha \log N}{1-\gamma} \ .$$

The inequality lets us compute the required N to make the looping policy superior with

$$N > \exp\big(R(1-\gamma)/(\alpha\gamma)\big) \ .$$

Comparing both stratgies leads to $J_\alpha(\pi_{\text{loop}}) > J_\alpha(\pi_{\text{exit}})$, so every soft-optimal policy puts its mass on the actions choosing the zero-reward loop, yielding $J(\pi_\alpha^{\text{SAC}}) = 0$ in the primal problem. The performance gap is therefore $J^\star - J(\pi_\alpha^{\text{SAC}}) = R = \varepsilon$, independent of α. Folding the entropy bonus into the reward does shift the critic target and it turns the primal optimization problem into a non-separable and thus non-convex linear program. The entropy bonus can thus drive the learned policy arbitrarily far from optimality and lead to degenerate solutions. Recovering an $O(\alpha)$ gap requires treating the KL/entropy term outside the reward or annealing $\alpha \to 0$.

Integrating entropy in the critic, as done, for example, in SAC, leads to a loss of theoretical near-optimality guarantees and can result in policies with unintended behavior.

We note that especially in sparse reward environments with small R over long periods of timesteps this condition is likely to be fulfilled, identifying a systematic weak spot and failure mode of SAC. Our algorithm fixes this by adhering to the original linear program and minimizes the near-optimality gap by using annealing on the target KL.

5 Kullback Leibler Regularized Actor Critic

We derive the general KL-regularized actor-critic algorithm and show that it retains near-optimality. We show that entropy regularization is a special case of our general KL-regularization. In the end, we formalize an easily applicable annealing scheme that generalizes across both continuous and discrete environments, having interpretable hyperparameters, and introduce a replacement for the entropy bonus with a uniform bias as a hyperparameter.

5.1 Constrained-Optimisation Derivation

We start from the γ-discounted control problem defined in Eq.(1). We then introduce the state-value multipliers $v \in \mathbb{R}^{|\mathcal{S}|}$, a scalar normaliser $\lambda \in \mathbb{R}$ as Lagrangian multiplier and the temperature $\alpha > 0$ for the KL in the lagrangian of the problem

$$\begin{aligned}\mathcal{L}(\mu, v, \alpha, \lambda) = &\sum_{s,a} r(s,a)\mu(s,a) - \alpha\left(\sum_{s,a}\mu(s,a)\log\frac{\mu(s,a)}{q(s,a)} - \varepsilon\right) \\ &- \sum_s v(s)\Big[\sum_a \mu(s,a) - \gamma\sum_{s',a} P(s'|s,a)\mu(s,a) - (1-\gamma)\mu(s)\Big] - \lambda\Big(\sum_{s,a}\mu(s,a) - 1\Big)\,.\end{aligned} \tag{5}$$

Since for fixed (v, λ, α) the Lagrangian in Eq.(5) is strictly concave in the occupancy measure μ, the inner maximization over μ is characterized by the Karush–Kuhn–Tucker stationarity condition $\partial\mathcal{L}/\partial\mu = 0$. Solving this first-order condition yields the Gibbs-form optimizer. We recover the actor and critic and derive their respective losses. First, we need to take the partial derivative with respect to the occupancy measure. Setting $\partial\mathcal{L}/\partial\mu = 0$ gives the Gibbs form

$$\mu^\star(s,a) = q(s,a)\,\exp\Big(\alpha^{-1}\big[A_v(s,a) - \lambda\big]\Big)\,, \quad A_v(s,a) = r(s,a) + \gamma\mathbb{E}_{s'}v(s') - v(s). \tag{6}$$

The resulting $\mu^\star$ induces the actor through its conditional action distribution, and substituting $\mu^\star$ back into Eq.(5) together with the normalization constraint $\sum_{s,a}\mu(s,a) = 1$ collapses the primal to the smooth convex dual

$$J_D(v) = (1-\gamma)\,\mathbb{E}_{s\sim\mu}v(s) + \frac{1}{\alpha}\log\sum_{s,a} q(s,a)\exp\big(\alpha A_v(s,a)\big)\,. \tag{7}$$

We define the Q function as the critic and the policy as a softmax, naturally spawning from the softmax interpretation of the critic given through the general KL constraint. This step has been proposed before (Peters et al., 2010), but we emphasize that, unlike previous work, we keep the constraint to a fixed distribution instead of the last policy. For any v, define the critic

$$Q_v(s,a) = r(s,a) + \gamma\mathbb{E}_{s'}v(s')\,,$$

and recover the actor by the Boltzmann projection

$$\pi_\alpha(a \mid s) = \frac{q(s,a)\exp\big(Q_v(s,a)/\alpha\big)}{\sum_b q(s,b)\exp\big(Q_v(s,b)/\alpha\big)} \ . \tag{8}$$

The Boltzmann projection makes explicit the dual role of the critic in shaping the policy while ensuring that the resulting actor remains consistent with the KL-regularized control objective.

5.2 Deriving Surrogate Losses used in Practice

Since we can not obtain the final policy through search methods such as done in (Peters et al., 2010) as we parametrize the critic and actor with deep neural networks, we need to employ gradient based optimization, which is possible due to the smooth convex dual structure. We derive the losses of the actor and critic and start with the Bellman error. We recall the dual objective of the general KL-regularisation written in terms of the network parameters θ and the lagrangian multiplier $\alpha > 0$:

$$g(\theta,\alpha) = \alpha \log \sum_{s,a} \exp\big(\delta_\theta(s,a)/\alpha\big) - \alpha\,\mathrm{KL}_{\min} \ , \quad \delta_\theta(s,a) = r(s,a) + \gamma\,\mathbb{E}_{s'} V_\theta(s') - Q_\theta(s,a) \ . \tag{9}$$

The quantity δ_θ is the Bellman residual, which is $\delta_\theta(s,a) = 0$ at the optimum for every state–action pair. The factor

$$Z(\delta) \ = \ \sum_{s,a} \exp\big(\delta_\theta(s,a)/\alpha\big)$$

is the partition function that guarantees $\sum_{s,a} \mu^\star(s,a) = 1$. For the exponential family this convex conjugate replacing the primal variables is the log-partition function

$$g(\theta,\alpha) = \alpha \log Z(\delta) - \alpha \mathrm{KL}_{\min} \ , \tag{10}$$

whose gradient is the softmax policy $\pi_\alpha = \nabla_\delta(\alpha \log Z)$ and whose Hessian is the Fisher information matrix so $\alpha \log Z$ acts as a natural potential for the saddle problem. The exact objective Wq. (10) is smooth and convex but expensive to minimize with deep networks due to the non-linearity. Therefore, we use a Taylor expansion of $\log Z$ around the optimum $Z(\delta) = \sum_{s,a} \exp(\delta/\alpha)$ and denote by $N = |\mathcal{S}||\mathcal{A}|$ the number of terms and by $\bar{\delta} = N^{-1} \sum_{s,a} \delta$ their empirical mean. When all residuals are small, we expand Z to second order

$$\begin{aligned} Z(\delta) &= N\Big[1 + \frac{\alpha}{N}\sum \delta + \frac{\alpha^2}{2N}\sum \delta^2 + O(\|\delta\|^3)\Big], \\ \log Z(\delta) &= \log N + \frac{\alpha}{N}\sum \delta + \frac{\alpha^2}{2N}\left[\sum \delta^2 - \frac{1}{N}\left(\sum \delta\right)^2\right] + O\big(\|\delta\|^3\big) \ . \end{aligned} \tag{11}$$

Because $\sum \delta^2 - N\bar{\delta}^2 = \sum(\delta - \bar{\delta})^2$ the quadratic term measures the variance of the residuals. Substituting Eq.(11) in Eq.(10) and retaining terms up to order δ^2 gives

$$g(\theta,\alpha) = \alpha \log N - \alpha \mathrm{KL}_{\min} + \frac{1}{N}\sum_{s,a} \delta_\theta(s,a) + \frac{1}{2\alpha N}\sum_{s,a} \big(\delta_\theta(s,a) - \bar{\delta}_\theta\big)^2 + O\big(\|\delta_\theta\|^3/\alpha^2\big) \ .$$

Close to the optimum, the linear term is negligible because $\sum \delta_\theta \to 0$. The leading contribution is therefore the variance-weighted quadratic term. Dropping constants and the cubic remainder yields the practical loss that drives the critic

$$\mathcal{L}_Q(\theta) = \frac{1}{2\alpha}\,\mathbb{E}_{(s,a)\sim w}\Big[\big(\delta_\theta(s,a) - \mathbb{E}_w[\delta_\theta]\big)^2\Big] \ ,$$

where w is the sampling distribution induced by the replay buffer. In practice, the scaling using the lagrangian multiplier α is omitted. Because the baseline $\mathbb{E}_w[\delta_\theta]$ does not depend on a, omitting it leaves the gradient

Algorithm 1 KL-Regularised Actor–Critic (KLAC) with Target-KL Annealing and Uniform Bias

Hyper-params: discount γ, learning rates $(\alpha_Q, \alpha_\pi, \alpha_\alpha)$, Polyak factor τ, bias β, exploit probabilities $(p_{\text{start}}, p_{\text{end}})$, anneal horizon T_{anneal}, half-width r (cont.)
Networks: critic Q_θ, target critic $Q_{\bar{\theta}} \leftarrow Q_\theta$, policy π_ϕ
Initialise temperature α_0, replay buffer $\mathcal{D}$, time step $t \leftarrow 0$
while training **do**
  Observe s_t
  Sample action $a_t \sim \pi_\phi(\cdot \mid s_t)$ Eq.(8)
  Execute a_t, receive (r_t, s_{t+1}), push (s_t, a_t, r_t, s_{t+1}) into $\mathcal{D}$
  if ready to update **then**
    Sample mini-batch $\{(s_i, a_i, r_i, s_i')\}_{i=1}^{B} \sim \mathcal{D}$
    $y_i \leftarrow r_i + \beta + \gamma\, \mathbb{E}_{a' \sim \pi_\phi(\cdot \mid s_i')} Q_{\bar{\theta}}(s_i', a')$
    $\mathcal{L}_Q \leftarrow \frac{1}{B} \sum_i \big(Q_\theta(s_i, a_i) - y_i\big)^2$ Eq.(12)
    $\theta \leftarrow \theta - \alpha_Q \nabla_\theta \mathcal{L}_Q$
    $\mathcal{L}_\pi \leftarrow \alpha_t \frac{1}{B} \sum_i D_{KL}\Big(\pi_\phi(\cdot \mid s_i) \,\Big\|\, \frac{\exp(Q_\theta(s_i, \cdot)/\alpha_t)}{Z_\theta(s_i)}\Big)$ Eq.(13)
    $\phi \leftarrow \phi - \alpha_\pi \nabla_\phi \mathcal{L}_\pi$
    $p_t \leftarrow p_{\text{start}} + \dfrac{t}{T_{\text{anneal}}} (p_{\text{end}} - p_{\text{start}})$
    if discrete action space **then**
      $\mathcal{H}_t^\star \leftarrow -p_t \log p_t - (1 - p_t) \log\big(\frac{1 - p_t}{|\mathcal{A}| - 1}\big)$
    else
      $\sigma_t \leftarrow \dfrac{r}{\sqrt{2}\ \mathrm{erf}^{-1}(p_t^{1/d})}, \quad d = |\mathcal{A}|$
      $\mathcal{H}_t^\star \leftarrow \frac{d}{2}(1 + \ln 2\pi) + d \ln \sigma_t$
    end if
    $\mathcal{L}_\alpha \leftarrow -\alpha_t \frac{1}{B} \sum_i \big(\mathcal{H}(\pi_\phi(\cdot \mid s_i)) - \mathcal{H}_t^\star\big)$ Eq.(10)
    $\alpha_{t+1} \leftarrow \alpha_t - \alpha_\alpha \nabla_{\alpha_t} \mathcal{L}_\alpha$
    $\bar{\theta} \leftarrow \tau\, \theta + (1 - \tau)\, \bar{\theta}$
  end if
  if s_{t+1} terminal **then**
    reset environment
  end if
  $t \leftarrow t + 1$
end while

unchanged, so $\mathcal{L}_Q$ reduces in practice to the mean-squared soft Bellman error used in Eq.(12). Using the network Q_θ and a replay buffer $\mathcal{D}$ we minimize

$$\mathcal{L}_Q(\theta) = \mathbb{E}_{(s,a,r,s') \sim \mathcal{D}} \Big[\big(Q_\theta(s, a) - r - \gamma\, \mathbb{E}_{a' \sim \pi_\alpha(\cdot \mid s')} Q_{\bar{\theta}}(s', a')\big)^2 \Big] . \tag{12}$$

Given Q_θ to obtain the actor we solve the convex projection

$$\mathcal{L}_\pi(\phi) = \alpha\, \mathbb{E}_{s \sim \mathcal{D}} \Big[\mathrm{D}_{\mathrm{KL}}\Big(\pi_\phi(\cdot \mid s) \,\Big\|\, \frac{\exp(q(s, \cdot) Q_\theta(s, \cdot)/\alpha)}{Z_\theta(s)}\Big) \Big] , \tag{13}$$

which gives us the loss function that we minimize using stochastic gradient descent.

5.3 Uniform Bias vs Entropy Bonus

In entropy-regularised actor–critic algorithms the critic target is shifted by the stochastic term $\alpha \mathcal{H}\big(\pi(\cdot \mid s')\big)$, a quantity that varies across states and training iterations and whose temperature α is notoriously difficult to tune. We argue that the empirical gains observed in some environments attributed to this entropy adjustment

can be captured equally well by a constant reward bias, which is far easier to interpret and gives practitioners an easily adjustable hyperparameter. It represents a fixed preference for exploratory actions, conceptually similar to epsilon-greedy algorithms, that keep the agent alive irrespective of state and is also known as a reward shaping method in the intrinsic motivation literature (Singh et al., 2010; Aubret et al., 2019). To our knowledge we are the first to highlight this connection between the success of SAC in some cases and the intrinsic motivation it actually uses, instead of the misattribution of entropy regularization of the critic. Concretely, replacing the entropy term in the critic target yields

$$y_t = r_t + \beta + \gamma\, Q_{\bar{\theta}}\big(s_{t+1}, a_{t+1}\big), \qquad a_{t+1} \sim \pi_\phi(\,\cdot \mid s_{t+1})\,,$$

so that the critic is updated via the usual squared-error loss $L(\theta) = \mathbb{E}[(Q_\theta(s_t, a_t) - y_t)^2]$. The constant β absorbs the average effect of $\alpha\mathcal{H}$ while avoiding its state-dependent fluctuations. Consequently, any performance gains previously ascribed to entropy shaping can be reinterpreted as arising from this fixed incentive term in the actor-critic framework.

5.4 Target-KL Annealing

Annealing is usually performed on the temperature parameter itself, adjusting the strength of the entropy bonus throughout training (Haarnoja et al., 2018). Target-entropy annealing, originally introduced for discrete action spaces (Xu et al., 2021a), instead changes the desired entropy level directly. In the discrete setting the target entropy can be interpreted as the average probability of exploiting the current best action. For a categorical policy with n actions we place a mass $p \in (0, 1)$ on the greedy action and spread the remainder evenly, which yields

$$\mathcal{H}_{\text{disc}}(p, n) = -p \log p - (1 - p) \log\!\Big(\tfrac{1-p}{n-1}\Big)\,.$$

We extend this intuition to continuous control by defining an exploitation region $\mathcal{E}_r = \{a \mid \|a - \mu\|_\infty \le r\}$ of half-width r around the policy mode μ. For the diagonal Gaussian policy used in SAC (Haarnoja et al., 2018) the probability of sampling within $\mathcal{E}_r$ is $p = [\operatorname{erf}(r/\sqrt{2}\sigma)]^d$, where d is the action dimension, which we invert to obtain the standard deviation

$$\sigma(p, r, d) = \tfrac{r}{\sqrt{2}\,\operatorname{erf}^{-1}(p^{1/d})}\,.$$

Substituting this into the entropy of a d-dimensional Gaussian,

$$\mathcal{H}_{\text{cont}}(p, r, d) = \frac{d}{2}\big(1 + \ln 2\pi\big) + d \ln \sigma(p, r, d)\,,$$

gives a closed-form mapping from any desired exploit probability p to a unique target entropy. Annealing $\mathcal{H}$ linearly between the values computed from $\{p_{\text{start}}, p_{\text{end}}\}$ therefore provides an interpretable, architecture-agnostic schedule that unifies target-entropy annealing across discrete and continuous action modalities while letting the practitioner specify the exploration–exploitation trade-off in intuitive probabilistic terms.

We derive KLAC, a KL-regularized actor–critic algorithm that generalizes entropy regularization, show that replacing entropy bonuses with a uniform bias yields a simpler, interpretable alternative, and introduce a probabilistic target-KL annealing scheme that generalizes across action spaces.

As a schedule, we chose a simple linear schedule as it worked well in our experiments, as shown later on. Further modifications for discrete settings, as done in (Xu et al., 2021b), might be beneficial in other environments. We show details of the complete algorithm KLAC in Algorithm 1.

6 Experiments

We will discuss experiments that highlight shortcomings of SAC, as described in the theory part, in practical applications. In addition, the experiments demonstrate the individual contributions of the concepts introduced in KLAC and its general performance. In the following, we will describe the implementation of SAC

Table 2: Index of the evaluated algorithms.

Label	Key idea
SAC	Actor regularized, entropy bias in critic, no annealing;
KLAC	Actor regularized with bias and annealing;
KLAC$_{-a}$	Actor regularized no annealing with bias;
KLAC$_{-b}$	Actor regularized with annealing no bias;
KLAC$_{-ab}$	Actor regularized no annealing no bias

and KLAC and three ablations of KLAC that we will evaluate. Experiments are performed on the MuJoCo continuous control (Todorov et al., 2012) and MinAtar discrete action (Young & Tian, 2019) benchmarks. We analyse five variants that differ only in their form of regularisation and reward bonus while sharing an identical implementation and hyperparameters. We compare our models to the original Soft Actor–Critic (SAC) with automatic temperature tuning (Haarnoja et al., 2018). Our default algorithm is **KLAC**, which uses the KL with uniform prior and bias term, and linearly decays the target KL. KLAC$_{-ab}$ removes the bias and annealing to assess the difference between using the entropy only in the actor and using it in both actor and critic as in SAC. In addition, we evaluate the individual impact of both the bias (KLAC$_{-a}$) and the annealing (KLAC$_{-b}$) by removing the respective other component. Table 2 shows the different variants.

6.1 Evaluation Criteria and Implementation Details

We use the implementation of cleanrl[1] (Huang et al., 2022) for SAC and as basis for KLAC and provide our own code for reference[2]. All hyperparameters are taken from the cleanrl implementation, except the replay buffer being adjusted to contain only 100000 steps in memory. In the discrete environments the agent employs a categorical policy whose logits are produced by a lightweight convolutional encoder (one 3×3 kernel with 16 feature maps followed by flattening, a 128–unit fully connected layer and a linear output head), while the twin soft-Q critics share the same encoder and replace the final head by value outputs, training uses Adam with decoupled learning rates of 3×10^{-4} for the actor and 3×10^{-4} for the critics, batches of 64 samples are drawn every four environment steps after a 20 000-step warm-up, target networks are updated every 8 000 steps, the base temperature α is learned online by gradient ascent on $\log\alpha$ to interpolate linearly from the exploitation target 0.50 to 0.80. Continuous-control experiments on MuJoCo use the standard twin MLP critics and a Tanh-Gaussian policy, each network comprising two 256–unit hidden layers, a replay buffer of 10^6 transitions, batches of 256, a 5 000-step warm-up, delayed policy updates every second critic step, and automatic entropy tuning with the target moving linearly. Continuous-control experiments are run on eight MuJoCo tasks [3]. Discrete-action results are collected on the five-game MinAtar suite Breakout-v1, Asterix-v1, Seaquest-v1, Freeway-v1, and SpaceInvaders-v1. Agents interact with the environment for one million steps on MuJoCo and three million steps on MinAtar. As aggregate measure we report the results with a mean, inter-quartile mean (IQM), and median plot (Agarwal et al., 2021). The IQM plot summarises learning curves by computing the inter-quartile mean of each algorithm's returns for the 10 best episodes overall. By discarding the top 25% and bottom 25% of scores before averaging, this robust aggregate retains the statistical efficiency of a mean while mitigating sensitivity to outliers and failed runs. The shaded regions in the plot show stratified-bootstrap 95% confidence intervals. The additional mean and median emphasize the statistical robustness of our results.

6.2 Actor Regularization is the Main Reason for good Performance of Adapted Actor-Critic Methods

In the first set of experiments we expect SAC to work well in environments with large and dense rewards. The MuJoCo test suite is a well-known benchmark, where SAC originally achieved superior performance and remains a state-of-the-art algorithm. In these well-performing environments, we investigate whether the

[1] https://github.com/vwxyzjn/cleanrl/tree/v1.0.0

[2] Link to repository in camera ready version

[3] Mujoco Environments: Hopper-v5, Walker2d-v5, HalfCheetah-v5, Ant-v5, Humanoid-v5, Swimmer-v5, InvertedPendulum-v5, Reacher-v5

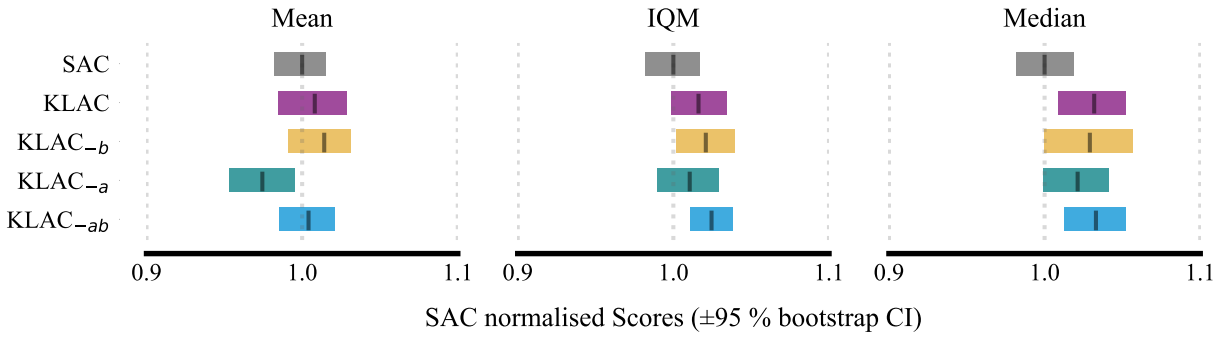

Figure 1: Mean, Inter-Quartile Mean (IQM), and Median for the algorithms SAC, KLAC, KLAC$_{-b}$ (without bias), KLAC$_{-a}$ (without annealing), and KLAC$_{-ab}$ (without bias and annealing) on 8 MuJoCo tasks. The scores for all algorithms are normalized using the SAC measure in the respective column. Confidence Intervals depict the 95% interval. The data is gathered by taking the 10 best episodes overall in each of the 20 runs of each algorithm during training until one million environment steps. In dense-reward MuJoCo tasks, all variants perform comparably. **Thus, the entropy bonus in the critic is irrelevant, and actor regularization alone drives performance.**

entropy bonus in the critic is the critical component as claimed (Haarnoja et al., 2018). Aggregate results for the continuous control environments are presented in Figure 1, computed from 20 independent training runs with different random seeds per environment for each algorithm variant. In terms of the best performance, measured by SAC-normalized scores, no method demonstrates a consistent superiority across environments. Furthermore, the close alignment of the mean, median, and interquartile mean (IQM) suggests that the performance distributions are relatively well-behaved, with minimal influence from extreme outliers. This consistency across summary statistics supports the robustness and reliability of the reported results. While KLAC does not provide a significant improvement over SAC, this is actually in line with our predictions. Dense reward environments with a reward scale that far eclipses the entropy bonus $\alpha H(\pi)$, like in MuJoCo, reinforce with similar results regardless of the entropy bonus, making it irrelevant. As SAC performs well in these environments, just like all KLAC variants, we show that the regularization of the actor is the driving force of the successful learning behavior.

The average exploration for SAC and KLAC is actually the same for methods with the same target entropy or target KL equivalent (see Appendix), further highlighting that the entropy bonus inside the critic does not influence entropy meaningfully in practice as results in rewards and entropy do not differ. Further, while our annealing adds more exploration by increasing variance, this does not yield higher rewards, as it is not needed in these dense reward environments. On the contrary, when investigating the entropy values and setting the target KL to be very high, decreasing the variance of the Gaussians, the resulting entropy values actually match the heuristic ones by SAC. The very low target entropy values of SAC show that SAC never does rich exploration (Figure 6 Appendix) and instead successfully exploits the dense reward signal even during training. The independence of the algorithm performance to an entropy signal and the heuristically low target entropy of SAC further highlight that it is not the entropy in the critic that drives improvement, but instead the regularization of the actor.

6.3 KLAC Prevents Critic Instability in Sparse Environments

In the next set of experiments we show that, while SAC performs competitively in many environments, its formulation allows the learned policy to drift arbitrarily far from the optimal solution. This tendency is particularly detrimental in environments with sparse rewards, small reward scales, or long horizons, where inaccurate value estimates can compound over time. In Figure 2, we observe this behaviour in the Asterix and Breakout task from MinAtar, where the average Q-values under SAC grow rapidly and far beyond the

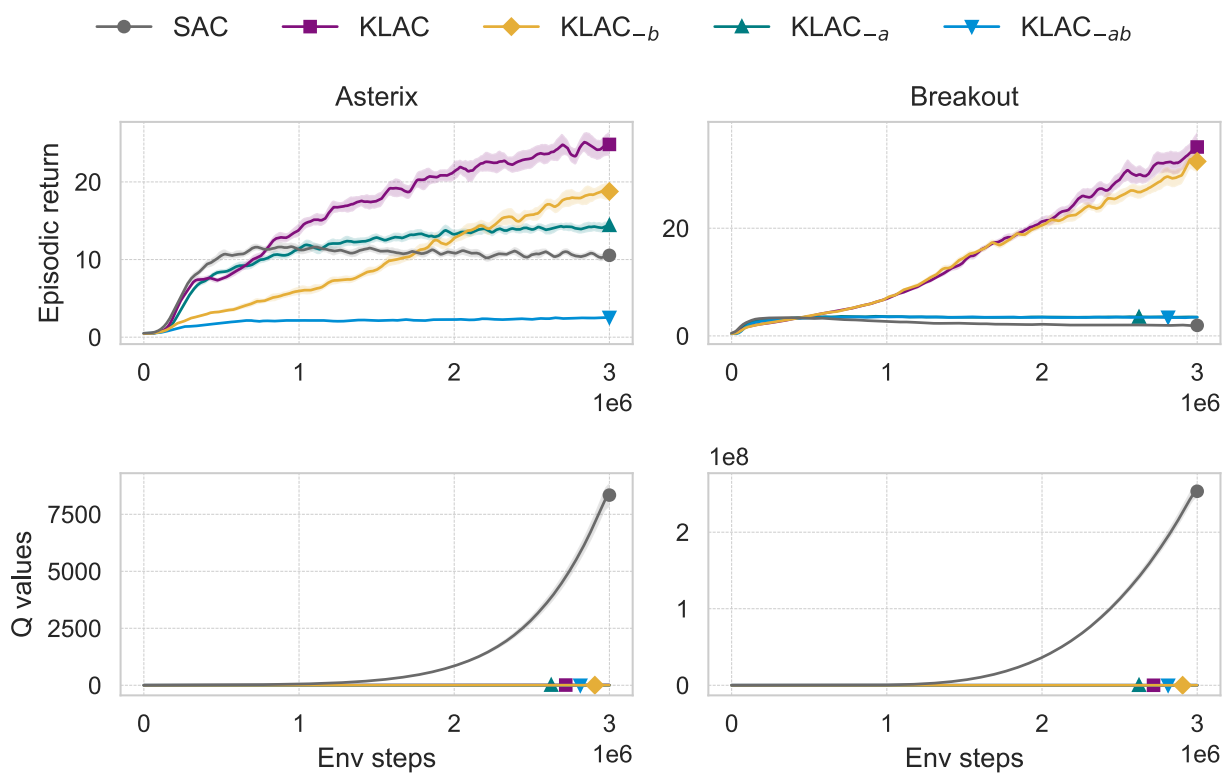

Figure 2: Episodic return (top row) and Q values (bottom row) for the algorithms SAC, KLAC, KLAC$_{-b}$ (without bias), KLAC$_{-a}$ (without annealing), and KLAC$_{-ab}$ (without bias and annealing) on 2 MinAtar tasks. The curves represent the smoothed mean at each time step across 20 random seeds with three million training steps. The shaded areas represent the 95% CI. **In sparse-reward MinAtar tasks, SAC suffers from uncontrolled Q-value growth and unstable performance, while KLAC stabilizes critics and achieves higher returns.**

scale justified by the observed rewards. This uncontrolled growth is accompanied by unstable policy updates and a degradation in asymptotic performance. Removing the entropy bonus in KLAC mitigates this effect, with the sole exception of Asterix, where we can see that an added bias to the reward obtains the same effect as the entropy. KLAC, using the bias and annealing, surpasses SAC in final returns, but we can see using KLAC$_{-b}$ that the bias plays an important role in Asterix. The additional optimism induced by entropy regularization accelerates early learning when dense rewards are present, representing one of the rare cases where SAC's bias is initially beneficial.

In environments with long horizons and sparse rewards, such as Seaquest (Figure 3), SAC's tendency to overestimate Q-values interacts with an implicit stay-alive bias, where the policy learns to prolong episodes without actively seeking rewards. In Asterix, this bias can occasionally support early learning by preserving opportunities for reward collection, but in Seaquest, it is harmful. The policy prioritises avoiding termination rather than discovering the sparse rewarding states, leading to stagnation at near-zero returns. The same effect can be observed for KLAC$_{-a}$, as the bias also makes surviving the dominant strategy, showing how the dual gap can get arbitrarily large from the primal, when one introduces custom terms into the reward function that do not align with the desired behavior. Moreover, in Seaquest, we observe a divergence between the critic's predictions and the actual returns collected. Even when the agent rarely encounters rewarding states, SAC's critic inflates value estimates over time (Figure 3), which disrupts the learning signal and prevents recovery without substantial corrective intervention. Removing the entropy bonus in KLAC prevents this

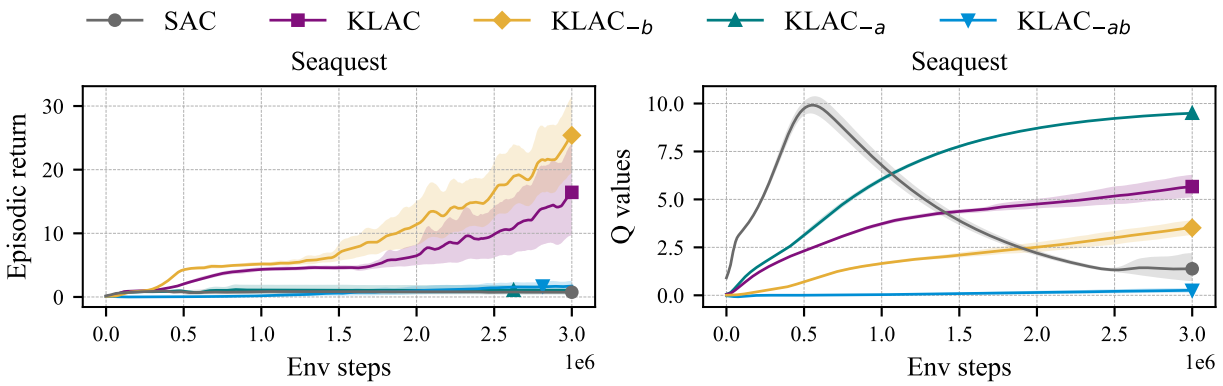

Figure 3: Episodic return (left) and Q values (right) for the algorithms SAC, KLAC, KLAC$_{-b}$ (without bias), KLAC$_{-a}$ (without annealing), and KLAC$_{-ab}$ (without bias and annealing) on Seaquest. The curves represent the smoothed mean at each time step across 20 random seeds with three million training steps. The shaded areas represent the 95% CI. **In long-horizon sparse tasks like Seaquest, SAC overestimates Q-values and stalls. KLAC prevents this divergence and yields consistently better learning.**

unchecked growth, restores correlation between Q-values and observed rewards, and consistently achieves better returns.

Annealing the target KL coefficient further improves final performance drastically on every discrete task while keeping variance low on every task. The progressive reduction of the target KL drives the policy towards lower entropy, thereby tightening the primal-dual gap and allowing the actor to converge to a more accurate estimate of the primal objective. These results are in line with our theoretical predictions and show that it is the reduction in entropy over time in the action distribution of the actor that is important to increase performance steadily over time. We note that simply starting with a very low KL target results in insufficient exploration, prohibiting the discovery of good policies consistently. Notably, KLAC is training faster on tasks and is not saturating at a premature convergence point (see Appendix). This leads us to the conclusion that annealing of the target KL, that is, relaxing the constraint over time by allowing larger deviations from the reference distribution instead of annealing the temperature parameter directly, is a crucial and necessary method to further enhance the performance of regularized policy search methods.

The aggregated performance profiles Agarwal et al. (2021) in Figure 4 confirm these environment-specific observations. Across all MinAtar games, KLAC outperforms SAC in all runs. In several tasks, KLAC achieves final returns exceeding SAC by a factor of up to 35, while avoiding the variance and premature saturation observed in the baseline. These results are consistent with our theoretical analysis. SAC's entropy-driven optimism can lead to severe overestimation in sparse or long-horizon tasks, producing ineffective exploration and degraded learning.

Our experiments confirm that actor regularization is the key factor behind strong performance in adapted actor–critic methods, not the entropy bonus in SAC's critic. KLAC prevents Q-value divergence in sparse tasks and, through uniform bias and target-KL annealing, achieves more stable and substantially higher returns across environments.

By replacing the entropy term with a tunable optimism bias and progressively annealing the target KL coefficient, KLAC produces stable value estimates, avoids pathological Q-value growth, and delivers consistently superior final performance without sacrificing early training speed.

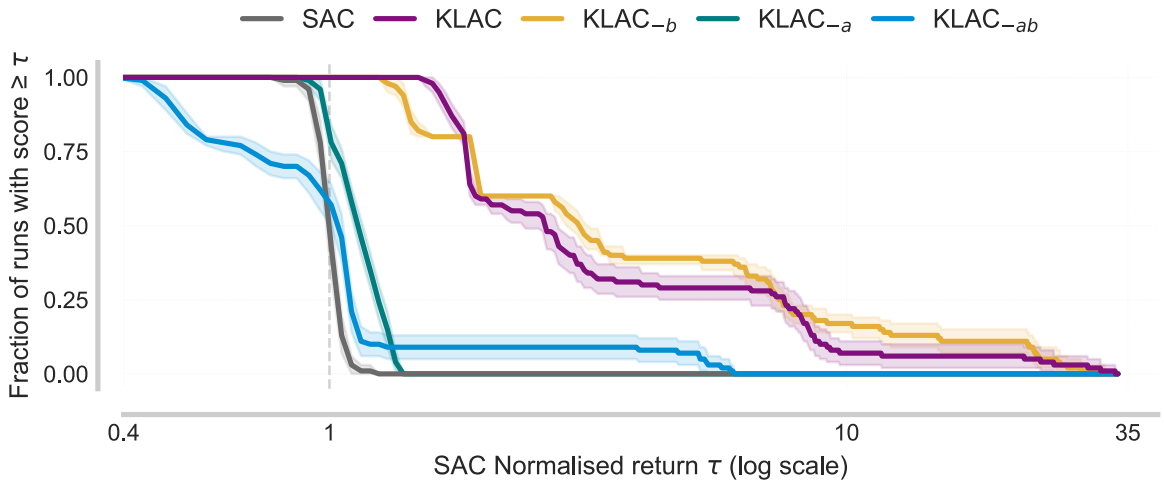

Figure 4: Performance profile plot for the algorithms SAC, KLAC, KLAC$_{-b}$ (without bias), KLAC$_{-a}$ (without annealing), and KLAC$_{-ab}$ (without bias and annealing). The plots are aggregated per environment over 20 runs each for each algorithm. The curve shows the proportion of runs on which each score falls within the multiplicative SAC mean performance of the respective environment. **Across all MinAtar environments, KLAC consistently dominates SAC, avoiding instability and achieving up to 35 times higher returns.**

7 Conclusion and Future Work

This work revisits maximum entropy reinforcement learning from first principles and reveals that the design of SAC, despite its empirical success, violates the theoretical foundations of regularized policy optimization. By adding entropy to the reward, SAC introduces a structural error that breaks convexity, prevents the application of standard duality guarantees, and can render the resulting policy arbitrarily suboptimal. We addressed this flaw by developing KLAC, an algorithm that restores theoretical soundness by regularizing explicitly through a KL divergence constraint. KLAC removes the entropy bonus from the critic, replaces it with an interpretable uniform bias, and introduces a target-KL annealing schedule that unifies discrete and continuous domains. Extensive experiments confirm that this design not only preserves SAC's empirical performance in dense-reward tasks but also prevents its instability in sparse environments, yielding stable and near-optimal solutions. Our findings invite a conceptual shift, rather than embedding entropy directly into the reward, entropy should be understood as one instance of KL regularization and applied as an explicit constraint. This reframing separates exploration incentives from value estimation, improves the interpretability of hyperparameters, and enables annealing schemes that translate naturally into probabilistic terms of exploitation.

Future work may extend KLAC in several directions. One promising avenue is its application to safety-critical domains and offline reinforcement learning, where bounded policy updates and trust regions are essential. Another direction is the exploration of general f-divergence regularizers within the KLAC framework, which may allow tailoring exploration strategies to specific environments. Additionally, investigating adaptive or state-dependent reference distributions could yield more efficient exploration while retaining convexity guarantees. Finally, large-scale empirical studies across domains with complex and deceptive reward structures would further validate KLAC's robustness and shed light on the practical trade-offs between bias, annealing, and constraint strength.

By disentangling principled regularization from heuristic reward shaping, KLAC demonstrates that reinforcement learning algorithms can be simultaneously empirically competitive and provably grounded, providing a foundation for more reliable and interpretable agents.

References

Amir Abdolmaleki, Maximilian Schwarzer, Jan Peters, and Gerhard Neumann. Maximum a posteriori policy optimization. In *International Conference on Learning Representations (ICLR)*, 2018.

Joshua Achiam, David Held, Aviv Tamar, and Pieter Abbeel. Constrained policy optimization. In *International conference on machine learning*, pp. 22–31. PMLR, 2017.

Rishabh Agarwal, Max Schwarzer, Pablo Samuel Castro, Aaron C Courville, and Marc Bellemare. Deep reinforcement learning at the edge of the statistical precipice. *Advances in neural information processing systems*, 34:29304–29320, 2021.

Eitan Altman. *Constrained Markov decision processes*. Routledge, 2021.

Argenis Arriojas, Jacob Adamczyk, Stas Tiomkin, and Rahul V Kulkarni. Entropy regularized reinforcement learning using large deviation theory. *Physical Review Research*, 5(2):023085, 2023.

Nair Ashvin, Dalal Murtaza, Gupta Abhishek, and L Sergey. Accelerating online reinforcement learning with offline datasets. *CoRR, vol. abs/2006.09359*, 2020.

Arthur Aubret, Laetitia Matignon, and Salima Hassas. A survey on intrinsic motivation in reinforcement learning. *arXiv preprint arXiv:1908.06976*, 2019.

Andrew G Barto. Reinforcement learning: An introduction. by richard's sutton. *SIAM Rev*, 6(2):423, 2021.

Richard Bellman. A markovian decision process. *Journal of mathematics and mechanics*, pp. 679–684, 1957.

Boris Belousov and Jan Peters. f-divergence constrained policy improvement. *arXiv preprint arXiv:1801.00056*, 2017.

Krzysztof Ciosek and Shimon Whiteson. Bias-variance trade-offs in deep reinforcement learning. *Journal of Machine Learning Research*, 20(128):1–31, 2019.

Christian Daniel, Gerhard Neumann, Oliver Kroemer, and Jan Peters. Hierarchical relative entropy policy search. *Journal of Machine Learning Research*, 17(93):1–50, 2016.

Matthieu Geist, Bruno Scherrer, and Olivier Pietquin. A theory of regularized markov decision processes. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97, pp. 2160–2169, 2019.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning*, pp. 1861–1870. Pmlr, 2018.

Jonathan Ho and Stefano Ermon. Generative adversarial imitation learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 4565–4573, 2016.

Shengyi Huang, Rousslan Fernand Julien Dossa, Chang Ye, Jeff Braga, Dipam Chakraborty, Kinal Mehta, and João G.M. Araújo. Cleanrl: High-quality single-file implementations of deep reinforcement learning algorithms. *Journal of Machine Learning Research*, 23(274):1–18, 2022. URL `http://jmlr.org/papers/v23/21-1342.html`.

V. K. Ivanov. On quasi-solutions of ill-posed problems. *Doklady Akademii Nauk SSSR*, 1962. Historical reference to Ivanov regularization.

Sham Kakade and John Langford. Approximately optimal approximate reinforcement learning. In *Proceedings of the 19th International Conference on Machine Learning (ICML)*, 2002.

Aviral Kumar, Justin Zhou, George Tucker, and Sergey Levine. Stabilizing off-policy q-learning via bootstrapping error accumulation reduction. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 11784–11794, 2019.

Sergey Levine. Reinforcement learning and control as probabilistic inference: Tutorial and review. *arXiv preprint*, arXiv:1805.00909, 2018.

Alan S Manne. Linear programming and sequential decisions. *Management Science*, 6(3):259–267, 1960.

Ofir Nachum, Yinlam Chow, and Mohammad Ghavamzadeh. Path consistency learning in tsallis entropy regularized mdps. *arXiv preprint arXiv:1802.03501*, 2018.

Gergely Neu, Anders Jonsson, and Vicenç Gómez. A unified view of entropy-regularized markov decision processes. *arXiv preprint arXiv:1705.07798*, 2017.

Luca Oneto, Sandro Ridella, and Davide Anguita. Tikhonov, ivanov and morozov regularization for support vector machine learning. *Machine Learning*, 103(1):103–136, 2016. doi: 10.1007/s10994-015-5540-x. Published online 22 December 2015.

Aldo Pacchiano, Jonathan N Lee, Peter Bartlett, and Ofir Nachum. Near optimal policy optimization via reps. *Advances in Neural Information Processing Systems*, 34:1100–1110, 2021a.

Aldo Pacchiano, Jonathan N. Lee, Peter L. Bartlett, and Ofir Nachum. Near optimal policy optimization via reps. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 34, pp. 1100–1110, 2021b.

Xia Pan, Ruiyi Zhang Liao, Le Wang, Ming Fei, and Yuan Yuan. Sd3: Stabilized deep deterministic policy gradients with delayed truncation. In *Proceedings of the 37th International Conference on Machine Learning (ICML)*, pp. 8079–8088, 2020.

Santiago Paternain, Luiz F O Chamon, Miguel Calvo-Fullana, and Alejandro Ribeiro. Constrained reinforcement learning has zero duality gap. In *Advances in Neural Information Processing Systems*, volume 32, pp. 7555–7565, 2019.

Jan Peters and Stefan Schaal. Natural actor-critic. *Neurocomputing*, 71(7-9):1180–1190, 2008.

Jan Peters, Katharina Mulling, and Yasemin Altun. Relative entropy policy search. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 24, pp. 1607–1612, 2010.

John Schulman, Sergey Levine, Philipp Moritz, Michael Jordan, and Pieter Abbeel. Trust region policy optimization. In *Proceedings of the 32nd International Conference on Machine Learning (ICML)*, pp. 1889–1897, 2015.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Satinder Singh, Richard L Lewis, Andrew G Barto, and Jonathan Sorg. Intrinsically motivated reinforcement learning: An evolutionary perspective. *IEEE Transactions on Autonomous Mental Development*, 2(2):70–82, 2010.

Olivier Terpin, Quentin Berthon, and Sébastien Coupet-Grimal. Optimal transport trust region policy optimization. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 23043–23053, 2022.

Andrey N. Tikhonov and Vasiliy Y. Arsenin. *Solutions of Ill-Posed Problems*. Scripta Series in Mathematics. V. H. Winston & Sons, Washington, D.C.; New York, 1977. ISBN 0-470-99124-0. Translated from the Russian; translation editor: Fritz John.

Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5026–5033. IEEE, 2012. doi: 10.1109/IROS.2012.6386109.

Nino Vieillard, Tadashi Kozuno, Bruno Scherrer, Olivier Pietquin, Rémi Munos, and Matthieu Geist. Leverage the average: An analysis of kl regularization in reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 33, 2020.

Oriol Vinyals, Igor Babuschkin, Wojciech Marian Czarnecki, Michaël Mathieu, Andrew Dudzik, Junyoung Chung, David H Choi, Richard Powell, Tobias Ewalds, Petko Georgiev, et al. Grandmaster level in starcraft ii using multi-agent reinforcement learning. *Nature*, 575(7782):350–354, 2019.

Tongzhou Wang and Bing Ni. Meta soft actor-critic for automatic temperature tuning. *arXiv preprint arXiv:2006.07390*, 2020.

Ziyu Wang, Tom Schaul, Matteo Hessel, Hado Van Hasselt, David Silver, and Thomas Degris. Sample efficient actor-critic with experience replay. In *International Conference on Learning Representations (ICLR)*, 2017.

Tong Wei, Zichuan Lin, Junliang Xing, Yuanchun Shi, Li Shen, Chao Yu, Deheng Ye, et al. Revisiting discrete soft actor-critic. *Transactions on Machine Learning Research*, 2025.

Ronald J Williams and Jing Peng. Function optimization using reinforcement learning. *Neural Computation*, 4(4):590–604, 1991.

Yifan Wu, George Tucker, and Ofir Nachum. Behavior regularized offline reinforcement learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 11720–11731, 2019.

Yuhuai Wu, John Schulman, Philipp Moritz, Pieter Abbeel, and Ilya Sutskever. Scalable trust-region method for deep reinforcement learning using kronecker-factored approximation. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 5279–5288, 2017.

Hexin Xu, Jing Fu, Shixiang Gu, and Sergey Levine. Discrete soft actor-critic with trust region constraint. In *International Conference on Learning Representations (ICLR)*, 2021a.

Yaosheng Xu, Dailin Hu, Litian Liang, Stephen McAleer, Pieter Abbeel, and Roy Fox. Target entropy annealing for discrete soft actor-critic. *arXiv preprint arXiv:2112.02852*, 2021b.

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

Haonan Yu, Haichao Zhang, and Wei Xu. Do you need the entropy reward (in practice)? *arXiv preprint*, arXiv:2201.12434, 2022a.

Haonan Yu, Haichao Zhang, and Wei Xu. Do you need the entropy reward (in practice)? *arXiv preprint arXiv:2201.12434*, 2022b.

Tom Zahavy, Brendan O'Donoghue, Guillaume Desjardins, and Satinder Singh. Reward is enough for convex mdps. In *Advances in Neural Information Processing Systems*, volume 34, pp. 25746–25759, 2021.

A Appendix

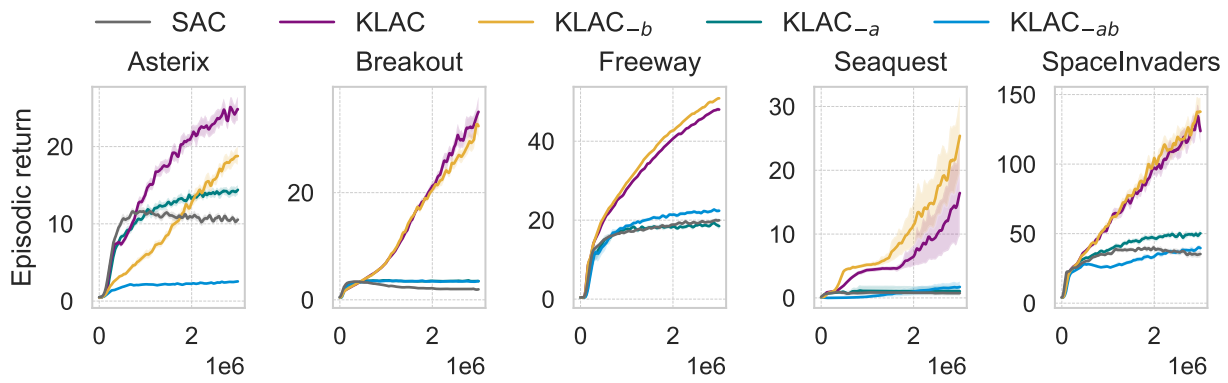

Figure 5: Episodic returns for the algorithms SAC, KLAC, KLAC$_{-b}$ (without bias), KLAC$_{-a}$ (without annealing), and KLAC$_{-ab}$ (without bias and annealing) on 5 MinAtar environments. The curves represent the mean at each time step for 20 runs with three million training steps. The shaded areas represent the standard deviation.

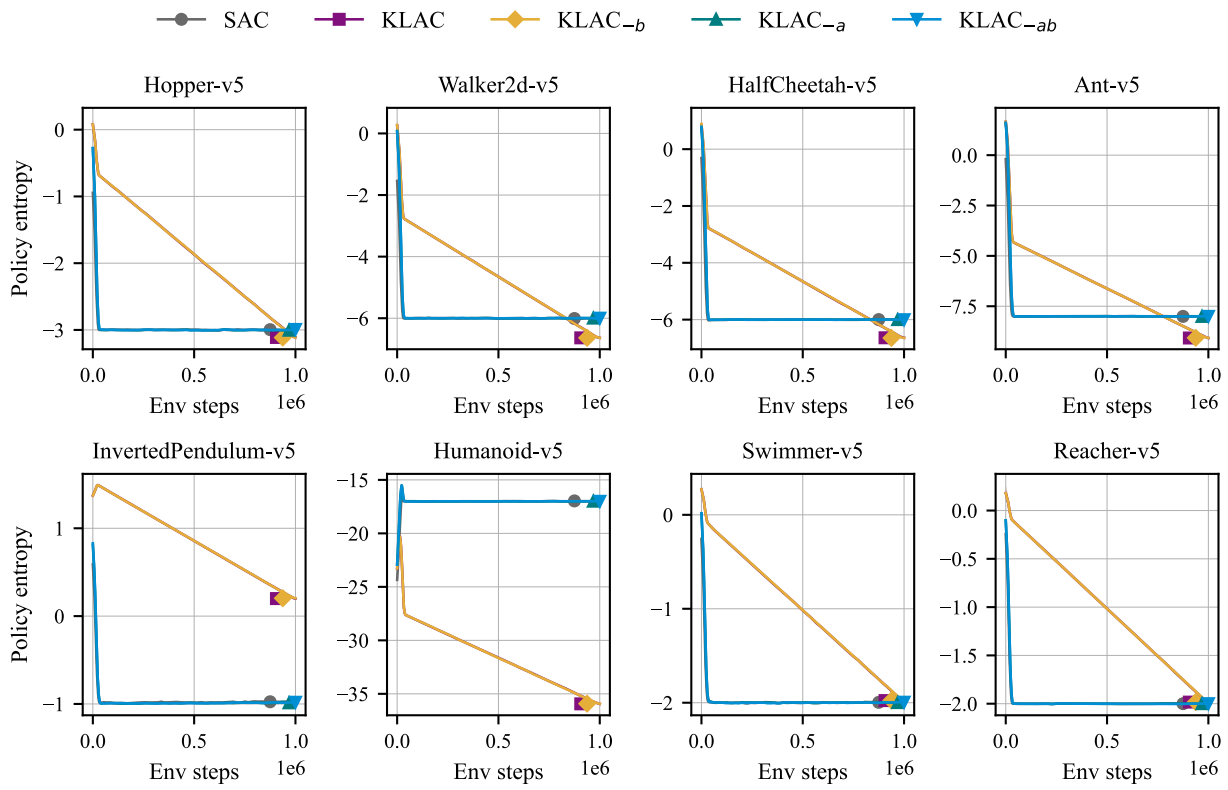

Figure 6: Mean entropy observed for the algorithms SAC, KLAC, KLAC$_{-b}$ (without bias), KLAC$_{-a}$ (without annealing), and KLAC$_{-ab}$ (without bias and annealing) on 8 MuJoCo environments. Each training step averaged over 20 runs in the Mujoco environments.