On Pathologies in KL-Regularized Reinforcement Learning from Expert Demonstrations

Anonymous Author(s)
Affiliation
Address
email

Abstract

KL-regularized reinforcement learning from expert demonstrations has proved successful in improving the sample efficiency of deep reinforcement learning algorithms, allowing them to be applied to challenging physical real-world tasks. However, we show that KL-regularized reinforcement learning with behavioral policies derived from expert demonstrations suffers from hitherto unrecognized pathological behavior that can lead to slow, unstable, and suboptimal online training. We show empirically that the pathology occurs for commonly chosen behavioral policy classes and demonstrate its impact on sample efficiency and online policy performance. Finally, we show that the pathology can be remedied by specifying *non-parametric* behavioral policies and that doing so allows KL-regularized RL to significantly outperform state-of-the-art approaches on a variety of challenging locomotion and dexterous hand manipulation tasks—without ad-hoc algorithmic design choices.

1 Introduction

Reinforcement learning [12, 18, 37, 38] is a powerful paradigm for learning complex behaviors. Unfortunately, many modern reinforcement learning algorithms require agents to carry out millions of interactions with their environment to learn desirable behaviors, making them of limited use for a wide range of practical applications that cannot be simulated [6, 22]. To improve the sample efficiency and performance of modern reinforcement learning algorithms, a significant body of prior work has explored approaches for efficiently incorporating expert demonstrations into the learning process [4, 8, 14, 33].

KL-regularized RL is a particularly successful approach for doing so [2, 21, 23, 25, 35, 42]. In KL-regularized RL, the standard reinforcement learning objective is augmented by a Kullback-Leibler (KL) divergence penalty that expresses the dissimilarity between the online policy and a behavioral reference policy derived from expert demonstrations. The resulting regularized objective pulls the agent's online policy towards the behavioral policy, while also allowing it to improve upon the behavioral policy by exploring and interacting with the environment. Recent advances that leverage explicit or implicit KL-regularized objectives, such as BRAC [42], ABM [35], and AWAC [21], have shown that KL-regularized RL from expert demonstrations is able to significantly improve the sample efficiency of online training and to solve challenging environments previously unsolved by standard deep RL algorithms.

Contributions. In this paper, we show that despite these successes, KL-regularized RL from expert demonstrations suffers from hitherto unrecognized pathologies that lead to instability and sub-optimality in online learning. Specifically, we demonstrate both theoretically and empirically that KL-regularized RL algorithms suffer from pathological behavior in online training when regularized against behavioral policies that experience a collapse in predictive variance about states away from

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

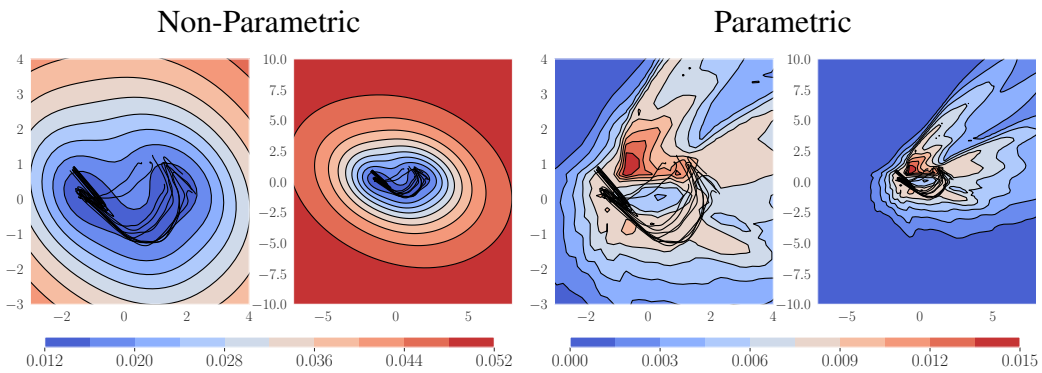


Figure 1: Predictive variances of non-parametric and parametric behavioral policies on a low dimensional representation (the first two principal components) of a 39-dimensional dexterous hand manipulation state space (see "door-binary-v0" in Figure 5). **Left**: Non-parametric Gaussian process posterior behavioral policy $\pi_{\mathcal{GP}}(\cdot \,|\, \mathbf{s}, \mathcal{D}_0) = \mathcal{GP}(\boldsymbol{\mu}_0(\mathbf{s}), \boldsymbol{\Sigma}_0(\mathbf{s}, \mathbf{s}'))$. **Right**: Parametric neural network Gaussian behavioral policy $\pi_\psi(\cdot \,|\, \mathbf{s}) = \mathcal{N}(\boldsymbol{\mu}_\psi(\mathbf{s}), \boldsymbol{\sigma}_\psi(\mathbf{s}))$. Expert trajectories $\mathcal{D}$ used to train the behavioral policies are shown in black. The GP predictive variance is well-calibrated: It is small near the expert trajectories and large in other parts of the state space. In contrast, the neural network predictive variance is poorly calibrated: It is relatively small on the expert trajectories, and collapses to near zero in unknown parts of the state space. Note the significant difference in scales.

the expert demonstrations. An example of such a collapse in predictive variance is shown in the right panel of Figure 1. We conduct a series of ablation studies that demonstrate that such pathological behavior occurs in commonly used parametric behavioral policy classes. Finally, we show that the pathology can be remedied by *non-parametric* behavioral policies, whose predictive variance is guaranteed not to collapse about previously unseen states, and that doing so results in online policies that significantly outperform state-of-the-art approaches on a range of challenging locomotion and dexterous hand manipulation tasks *without* ad-hoc design choices. The left panel of Figure 1 shows the predictive variance of a non-parametric behavioral policy, which—unlike in the case of the parametric policy shown in the right panel—increases off the expert trajectories. By avoiding the pathology, we obtain a stable and reliable approach to sample-efficient reinforcement learning, applicable to a wide range of reinforcement learning algorithms that leverage KL-regularized objectives. Visualizations of our results can be found at `https://sites.google.com/view/nppac`.

2 Background

We consider the standard reinforcement learning setting where an agent interacts with a discounted Markov Decision Process (MDP) [37] given by the 5-tuple $(\mathcal{S}, \mathcal{A}, p, r, \gamma)$, where $\mathcal{S}$ and $\mathcal{A}$ are the state and action spaces, $p(\cdot \,|\, \mathbf{s}_t, \mathbf{a}_t)$ are the transition dynamics, $r(\mathbf{s}_t, \mathbf{a}_t)$ is the reward function, and γ is a discount factor. $\rho_\pi(\boldsymbol{\tau}_t)$ denotes the state–action trajectory distribution from time t induced by a policy $\pi(\mathbf{a}_t \,|\, \mathbf{s}_t)$. The discounted return from time step t is given by $R(\boldsymbol{\tau}_t) = \sum_{k=t}^{\infty} \gamma^k r(\mathbf{s}_k, \mathbf{a}_k)$ for $t \in \mathbb{N}_0$. The standard reinforcement learning objective to be maximized is the expected discounted return $J_\pi(\boldsymbol{\tau}_0) = \mathbb{E}_{\rho_\pi(\boldsymbol{\tau}_0)}[R(\boldsymbol{\tau}_0)]$ under the policy trajectory distribution.

2.1 Improving & Accelerating Online Training via Behavioral Cloning

We consider settings where we have a set of expert demonstrations *without reward*, $\mathcal{D}_0 = \{(\mathbf{s}_n, \mathbf{a}_n)\}_{n=1}^N = \{\bar{\mathbf{S}}, \bar{\mathbf{A}}\}$, which we would like to use to speed up and improve online learning [4, 33]. One approach for turning the expert trajectories into a policy is behavioral cloning [1, 3] which involves learning a mapping from states in the expert demonstrations to their corresponding actions, that is, $\pi_0 : \mathcal{S} \to \mathcal{A}$. As such, behavioral cloning does not assume or require access to a reward function and only involves learning a mapping from states to action in a supervised fashion.

Since expert demonstrations are costly to obtain and often only available in small number, behavioral cloning alone is typically insufficient for agents to learn good policies in complex environments and has to be complemented by a method that enables the learner to build on the cloned behavior by interacting with the environment. A particularly successful and popular class of algorithms used for incorporating behavioral policies into online training is KL-regularized reinforcement learning [7, 31, 34, 39].

2.2 KL-Regularized Objectives in Reinforcement Learning

KL-regularized RL modifies the standard reinforcement learning objective by augmenting the return with a negative Kullback-Leibler (KL) divergence term from the learned policy π to a reference policy π_0, given a temperature parameter α. The resulting discounted return from time step $t \in \mathbb{N}_0$ is then given by

$$\tilde{R}(\boldsymbol{\tau}_t) = \sum_{k=t}^{\infty} \gamma^k \big[r(\mathbf{s}_k, \mathbf{a}_k) - \alpha \mathbb{D}_{\mathrm{KL}}(\pi(\cdot \,|\, \mathbf{s}_k) \parallel \pi_0(\cdot \,|\, \mathbf{s}_k)) \big] \tag{1}$$

and the reinforcement learning objective becomes $\tilde{J}_\pi(\boldsymbol{\tau}_0) = \mathbb{E}_{\rho_\pi(\boldsymbol{\tau}_0)}[\tilde{R}(\boldsymbol{\tau}_0)]$. When the reference policy π_0 is given by a uniform distribution, we recover the entropy-regularized reinforcement learning objective used in Soft Actor–Critic (SAC) [10] up to an additive constant.

Under a uniform reference policy π_0, the resulting objective encourages exploration, while also choosing high-reward actions. In contrast, when π_0 is non-uniform, the agent is discouraged to explore areas of the state space $\mathcal{S}$ where the variance of $\pi_0(\cdot \,|\, \mathbf{s})$ is low (that is, more certain) and encouraged to explore areas of the state space where the variance of $\pi_0(\cdot \,|\, \mathbf{s})$ is high. The KL-regularized reinforcement learning objective can be optimized via policy–gradient and actor–critic algorithms.

2.3 KL-regularized Actor–Critic

An optimal policy π that maximizes the expected KL-augmented discounted return $\tilde{J}_\pi$ can be learned by directly optimizing the policy gradient $\nabla_\pi \tilde{J}_\pi$. However, this policy gradient estimator exhibits high variance, which can lead to unstable learning. Actor–critic algorithms [5, 13, 26, 32] attempt to reduce this variance by making use of the value function $V^\pi(\mathbf{s}_t) = \mathbb{E}_{\rho_\pi(\boldsymbol{\tau}_t)}[\tilde{R}(\boldsymbol{\tau}_t) \,|\, \mathbf{s}_t]$ or the action–value function $Q^\pi(\mathbf{s}_t, \mathbf{a}_t) = \mathbb{E}_{\rho_\pi(\boldsymbol{\tau}_t)}[\tilde{R}(\boldsymbol{\tau}_t) \,|\, \mathbf{s}_t, \mathbf{a}_t]$ to stabilize training.

Given a reference policy $\pi_0(\mathbf{a}_t \,|\, \mathbf{s}_t)$, the state value function can be shown to satisfy the modified Bellman equation

$$V^\pi(\mathbf{s}_t) \doteq \mathbb{E}_{\mathbf{a}_t \sim \pi}[Q^\pi(\mathbf{s}_t, \mathbf{a}_t)] - \alpha \mathbb{D}_{\mathrm{KL}}\big(\pi(\cdot \mid \mathbf{s}_t) \,||\, \pi_0(\cdot \,|\, \mathbf{s}_t)\big)$$

with a recursively defined Q-function

$$Q^\pi(\mathbf{s}_t, \mathbf{a}_t) \doteq r(\mathbf{s}_t, \mathbf{a}_t) + \gamma \, \mathbb{E}_{\mathbf{s}_{t+1} \sim p}[V^\pi(\mathbf{s}_{t+1})].$$

Instead of directly optimizing the objective function $\tilde{J}_\pi$ via the policy gradient, actor–critic methods alternate between policy evaluation and policy improvement [5, 10]:

Policy Evaluation. During the policy evaluation step, $Q^\pi_\theta(\mathbf{s}, \mathbf{a})$, parameterized by parameters θ is trained by minimizing the Bellman residual

$$J_Q(\theta) \doteq \mathbb{E}_{(\mathbf{s}_t, \mathbf{a}_t) \sim \mathcal{D}} \Big[\big(Q_\theta(\mathbf{s}_t, \mathbf{a}_t) - (r(\mathbf{s}_t, \mathbf{a}_t) + \gamma \mathbb{E}_{\mathbf{s}_{t+1} \sim p}[V_{\bar{\theta}}(\mathbf{s}_{t+1})]) \big)^2 \Big], \tag{2}$$

where $\mathcal{D}$ is a replay buffer and $\bar{\theta}$ is a stabilizing moving average of parameters.

Policy Improvement. In the policy improvement step, the policy π_ϕ, parameterized by parameters ϕ, is updated towards the exponential of the KL-augmented Q-function,

$$J_\pi(\phi) \doteq \mathbb{E}_{\mathbf{s}_t \sim \mathcal{D}} \left[\alpha \mathbb{D}_{\mathrm{KL}}(\pi_\phi(\cdot \,|\, \mathbf{s}_t) \parallel \pi_0(\cdot \,|\, \mathbf{s}_t)) \right] - \mathbb{E}_{\mathbf{s}_t \sim \mathcal{D}} \left[\mathbb{E}_{\mathbf{a}_t \sim \pi_\phi} \left[Q_\theta(\mathbf{s}_t, \mathbf{a}_t) \right] \right], \tag{3}$$

with states sampled from a replay buffer $\mathcal{D}$ and actions sampled from the parameterized online policy π_ϕ. The following sections will focus on the policy improvement objective and how certain types of references policies can lead to pathologies when optimizing $J_\pi(\phi)$ with respect to ϕ.

3 Identifying the Pathology

In this section, we investigate the effect of KL-regularization on the training dynamics. To do so, first, we carefully consider the properties of the KL divergence to identify a potential failure mode for KL-regularized reinforcement learning. Next, we consider parametric Gaussian behavioral policies commonly used in practice for continuous control tasks [10, 42] and show that for Gaussian behavioral policies with small predictive variance, the policy improvement objective suffers from exploding gradients with respect to the policy parameters ϕ. We confirm that this failure occurs empirically and demonstrate that it results in slow, unstable, and suboptimal online training.

3.1 When are KL-Regularized RL Objectives Meaningful?

We start by considering the properties of the KL divergence and discuss how these properties can lead to potential failure modes in KL-regularized objectives. A well known property of KL-regularized objectives in the variational inference literature is the occurrence of singularities when one distribution loses effective support of the other.

To illustrate this problem, we consider the case of Gaussian behavioral and learned policies, as are often chosen in practice. In theory, the KL divergence between two full Gaussian distributions is always finite and well-defined. Hence, we might hope KL-regularized RL with Gaussian behavioral and online policies to be unaffected by the failure mode described above. However, a Gaussian online policy $\pi_\phi(\cdot \mid \mathbf{s}_t)$ loses effective support with $\pi_0(\cdot \mid \mathbf{s}_t)$ as the predictive variance $\boldsymbol{\sigma}_0^2$ tends to zero, and hence: $\mathbb{D}_{\text{KL}}(\pi_\phi(\cdot \mid \mathbf{s}_t) \parallel \pi_0(\cdot \mid \mathbf{s}_t)) \to \infty$ as $\boldsymbol{\sigma}_0^2 \to 0$. In other words, in the limit of zero variance in the behavioral policy, the distribution becomes degenerate and will cause the KL divergence to be infinite [19]. While in practice, no behavioral policy will operate in the limit of zero variance, the functional form of the KL divergence between (univariate) Gaussians,

$$\mathbb{D}_{\text{KL}}(\pi_\phi(\cdot \mid \mathbf{s}_t) \parallel \pi_0(\cdot \mid \mathbf{s}_t)) \propto \log \frac{\boldsymbol{\sigma}_0}{\boldsymbol{\sigma}_\phi} + \frac{\boldsymbol{\sigma}_\phi^2 + (\boldsymbol{\mu}_\phi - \boldsymbol{\mu}_0)^2}{2\boldsymbol{\sigma}_0^2},$$

implies a continuous, quadratic increase in the magnitude of the divergence as $\boldsymbol{\sigma}$ decreases, further exacerbated by a large difference in predictive means, $|\boldsymbol{\mu}_\phi - \boldsymbol{\mu}_0|$.

As a result, for a Gaussian behavioral policy $\pi_0(\cdot \mid \mathbf{s}_t)$ with very small variance ($\boldsymbol{\sigma}_0^2 < 10^{-2}$) that places very low probability on points in sample space far away from the distribution's mean $\boldsymbol{\mu}_0$, computing the KL divergence can result divergence values so large to cause numerical instabilities and arithmetic overflow. Hence, even for a suitably chosen behavioral policy class, we see that a small predictive variance in the behavioral policy—which causes the learned policy to "essentially" lose support with respect to the behavioral policy—can result in the KL divergence blowing up at evaluation points not close to the behavioral policy's predictive mean.

One way of remedying this failure mode in practice would be to constrain the output of the variance network (e.g., by adding a small constant bias). However, to encourage effectively learning guaranteeing a flat, non-zero predictive variance far from the offline dataset is not enough. Rather, we require calibrated predictive variance estimates that increase on data points away from the expert demonstrations and can thus reduce the regularization towards poor predictive means on such points. Another solution could be to use distributions with heavier tails, e.g., a Laplace distribution over a Gaussian to avoid the quadratic scaling on the difference of means. However, we show in Appendix K that doing so does not fully resolve the pathology.

In the following sections, we explain how an explosion in $\mathbb{D}_{\text{KL}}(\pi_\phi(\cdot \mid \mathbf{s}_t) \parallel \pi_0(\cdot \mid \mathbf{s}_t))$ caused by small $\boldsymbol{\sigma}_0^2$ affects the gradients of $J_\pi(\phi)$ in KL-regularized RL and discuss of how and why $\boldsymbol{\sigma}_0^2$ may tend to zero in practice.

3.2 Exploding Gradients in KL-Regularized Reinforcement Learning Objectives

To understand how small predictive variances in behavioral policies can affect—and possibly destabilize—online training in KL-regularized RL, we consider the contribution of the behavioral policy's variance to the gradient of the policy objective in Equation (3). Compared to entropy-regularized actor–critic methods (SAC, Haarnoja et al. [10]), which implicitly regularize against a uniform policy, the gradient estimator $\hat{\nabla}_\phi J_\pi(\phi)$ in KL-regularized RL gains an extra scaling term $\nabla_{\mathbf{a}_t} \log \pi_0(\mathbf{a}_t \mid \mathbf{s}_t)$, the gradient of the prior log-density evaluated actions $\mathbf{a}_t$ sampled from the online policy:

Proposition 1 (Exploding Gradients in KL-Regularized RL)**.** *Let $\pi_0(\cdot \mid \mathbf{s})$ be a Gaussian behavioral policy with mean $\boldsymbol{\mu}_0$ and variance $\boldsymbol{\sigma}_0^2$, and let $\pi_\phi(\cdot \mid \mathbf{s})$ be an online policy with reparameterization $\mathbf{a}_t = f_\phi(\epsilon_t; \mathbf{s}_t)$ and random vector ϵ_t. The gradient of the policy loss with respect to the online policy's parameters ϕ is then given by*

$$\begin{aligned}\hat{\nabla}_\phi J_\pi(\phi) = \big(&\alpha \nabla_{\mathbf{a}_t} \log \pi_\phi(\mathbf{a}_t \mid \mathbf{s}_t) - \alpha \nabla_{\mathbf{a}_t} \log \pi_0(\mathbf{a}_t \mid \mathbf{s}_t) \\ &- \nabla_{\mathbf{a}_t} Q(\mathbf{s}_t, \mathbf{a}_t)\big) \nabla_\phi f_\phi(\epsilon_t; \mathbf{s}_t) + \alpha \nabla_\phi \log \pi_\phi(\mathbf{a}_t \mid \mathbf{s}_t)\end{aligned} \tag{4}$$

with $\nabla_{\mathbf{a}_t} \log \pi_0(\mathbf{a}_t \mid \mathbf{s}_t) = -\frac{\mathbf{a}_t - \boldsymbol{\mu}_0}{\boldsymbol{\sigma}_0^2}$. For fixed $|\mathbf{a}_t - \boldsymbol{\mu}_0|$, $\nabla_{\mathbf{a}_t} \log \pi_0(\mathbf{a}_t | \mathbf{s}_t)$ grows as $\mathcal{O}(\boldsymbol{\sigma}_0^{-2})$; thus,

$$|\hat{\nabla}_\phi J_\pi(\phi)| \to \infty \quad \textit{as} \quad \boldsymbol{\sigma}_0^2 \to 0 \quad \textit{whenever} \quad \nabla_\phi f_\phi(\epsilon_t; \mathbf{s}_t) \neq 0.$$

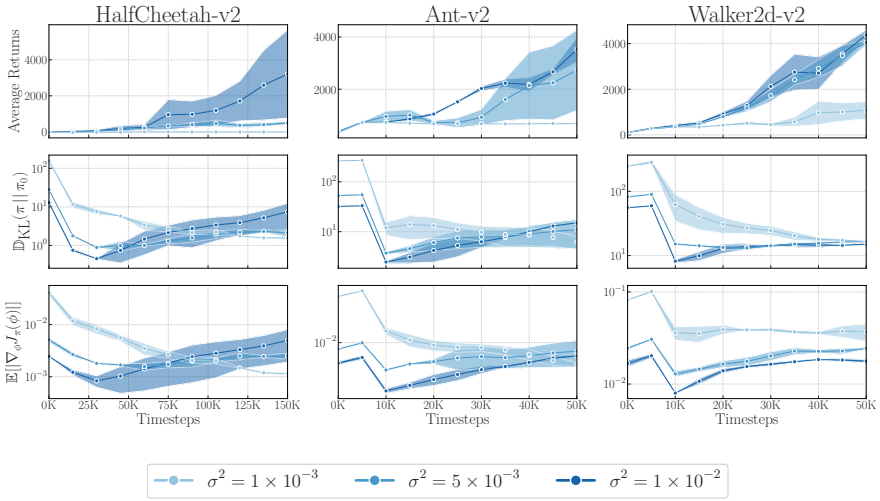


Figure 2: Ablation study showing the effect of predictive variance collapse on the performance of KL-regularized RL on MuJoCo benchmarks. The plots show the average return of the learned policy, magnitude of the KL penalty, and magnitude of the average absolute gradients of the policy loss during online training. The lighter the shading, the lower the behavioral policy's predictive variance.

Proof. See Appendix A. □

This result formalizes the intuition presented in Section 3.1 that a behavioral policy with a sufficiently small predictive variance may cause KL-regularized RL to suffer from pathological behavior in gradient-based optimization. The smaller the behavioral policy's predictive variance, the more sensitive the policy objective's gradients will be to differences in the means of the online and behavioral policies. As a result, for behavioral policies with small predictive variance, the KL divergence will heavily penalize online policies whose predictive means diverge from the predictive mean of the behavioral policy—even at states off the expert trajectory where the behavioral policy's mean prediction is poor.

3.3 Predictive Uncertainty Collapse under Parametric Policies

The most commonly used method for training behaviorally cloned stochastic policies is maximum likelihood estimation [42, 35], where we seek

$$\pi_0 \doteq \pi_{\psi^\star} \quad \text{with} \quad \psi^\star \doteq \arg\max_{\psi} \left\{ \mathbb{E}_{(\mathbf{s},\mathbf{a})\sim\mathcal{D}_0}[\log \pi_\psi(\mathbf{a}\,|\,\mathbf{s})] \right\}.$$

For commonly used Gaussian behavioral policies $\pi_\psi(\mathbf{a}\,|\,\mathbf{s}) = \mathcal{N}(\boldsymbol{\mu}_\psi(\mathbf{s}), \boldsymbol{\sigma}_\psi(\mathbf{s}))$ with $\boldsymbol{\mu}_\psi(\mathbf{s})$ and $\boldsymbol{\sigma}_\psi(\mathbf{s}))$ parameterized by a neural network, the maximum-likelihood loss on $\mathcal{D}_0 = \{(\mathbf{s}_n, \mathbf{a}_n)\}_{n=1}^N = \{\bar{\mathbf{S}}, \bar{\mathbf{A}}\}$ is given by $-\log \pi_\psi(\bar{\mathbf{A}} \mid \bar{\mathbf{S}}) = \sum_{n=1}^N \frac{\log \boldsymbol{\sigma}^2_\psi(\mathbf{s}_n)}{2} + \frac{(\mathbf{a}_n - \boldsymbol{\mu}_\psi(\mathbf{s}_n))^2}{2\boldsymbol{\sigma}^2_\psi(\mathbf{s}_n)} + C$, where C is a constant.

While maximizing the likelihood of the expert trajectories under the behavioral policy is a sensible choice for behavioral cloning, the limited capacity of the neural network parameterization can produce unwanted properties in the resulting policy. The maximum likelihood objective ensures that the behavioral policy's predictive mean reflects the expert's actions and the predictive variance the (aleatoric) uncertainty inherent in the expert trajectories.

However, even under additional regularization, the predictive variance of the resulting behavioral policy will invariably collapse to near-zero off the expert trajectory data manifold, since the parametric nature of the model limits the set of stochastic functions that can be represented by the policy. As a result, the set of possible action realizations attainable from a parametric behavioral policy only covers a small fraction of the action space, since the model's capacity is "used up" by fitting to the expert trajectories and since it is not encouraged to place higher predictive variance away from the training data. This behavior is well-known in parametric probabilistic models and well-documented in the approximate Bayesian inference literature [27].

Figure 1 demonstrates the collapse in predictive variance under maximum likelihood estimation in a low-dimensional representation of the "door-binary-v0" dexterous hand manipulation environment. It shows that while the predictive variance is small close to the expert trajectories (depicted as black lines), it rapidly decreases further away from them. Examples of variance collapse in other environments are presented in Appendix E. Figure 3 shows that the predictive variance off the expert trajectories consistently decreases during training over multiple random seeds. As shown in Proposition 1, such a collapse in predictive variance can result in pathological behavior in KL-regularized online training, since it steers the online policy towards suboptimal trajectories off the offline data manifold and results in suboptimal performance.

Alternative Behavioral Policy Regularization Schemes To prevent a collapse in the behavioral policy's predictive variance, prior work proposed adding entropy or Tikhonov regularization to the MLE objective [42]. However, doing so does not achieve a collapse in predictive variance off the expert demonstration trajectories, as we show in Appendix C. A widely used method to obtain predictive uncertainty estimates from neural networks in regression settings are probabilistic ensembles [16], where the predictive mean and variance are estimates from the predictive means and variances of multiple Gaussian neural networks. However, this approach is costly, as it requires training multiple neural networks from scratch and does not guarantee well-calibrated uncertainty estimates [40].

3.4 Empirical Confirmation of Uncertainty Collapse

To confirm Proposition 1 empirically and assess the effect of the collapse in predictive variance on the performance of KL-regularized RL, we perform an ablation study where we fix a behavioral policy's predictive mean function and vary the magnitude of the policy's predictive variance. The predictive mean function is chosen to attain 60% of the optimal performance and the behavioral policy's predictive variance is set to a constant value from the set $\{1 \times 10^{-3}, 5 \times 10^{-3}, 1 \times 10^{-2}\}$ (following a similar implementation in Nair et al. [21]). [1] The results are shown in Figure 2, which plots the average returns, the KL divergence, and the average absolute gradients of the policy loss over training. They confirm that as the predictive variance of the offline behavioral policy tends to zero, the KL terms and average policy gradient magnitude explode as hypothesized, leading to unstable training and a collapse or dampening in average returns.

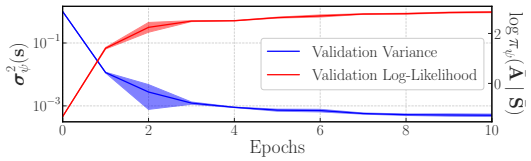


Figure 3: Collapse in the predictive variance (in **blue**) of a Gaussian behavioral policy parameterized by a neural network when training via maximum likelihood estimation. Lines and shaded regions denote means and standard deviations over five random seeds, respectively.

In other words, even for behavioral policies with accurate predictive means, smaller predictive variances slow down or even entirely prevent learning good behavioral policies. This observation confirms that the pathology identified in Proposition 1 occurs in practice and that it can have a significant impact on KL-regularized RL from expert demonstrations, calling into question the usefulness of KL regularization as a means for accelerating and improving online training. In Appendix F, we show that an analogous relationship exists for the gradients of the Q-function loss.

4 Fixing the Pathology

In order to address the collapse in predictive uncertainty for behavioral policies parameterized by a neural network trained via MLE, we specify a *non-parametric* distribution over behavioral policies whose predictive variance is *guaranteed* not to collapse about previously unseen states and propose N-PPAC: Non-Parametric Prior Actor–Critic.

4.1 Non-Degenerate Gaussian Processes

Gaussian processes (GPs) [30] are models over functions defined by a mean $m(\cdot)$ and covariance function $k(\cdot, \cdot)$. When defined in terms of a non-parametric covariance function, that is, a covariance function constructed from infinitely many basis functions, we obtain a non-degenerate GP, which has

[1] We note that smaller values resulted in arithmetic overflow.

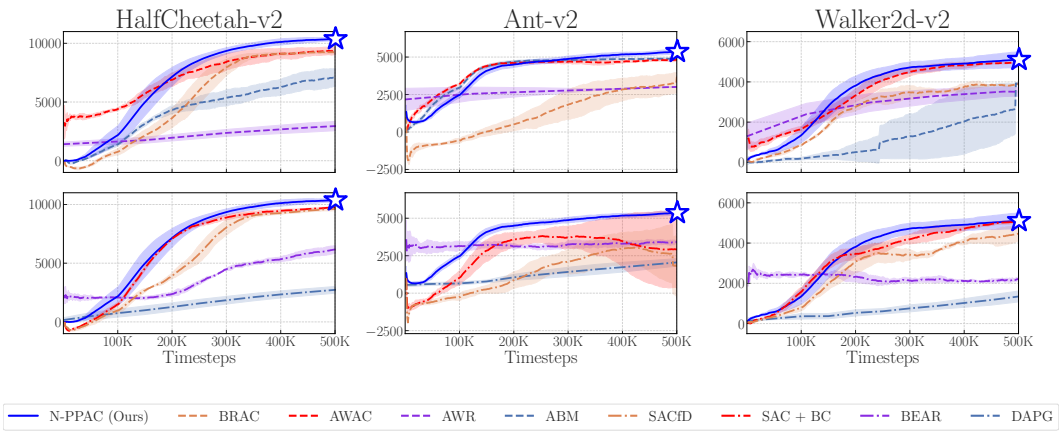


Figure 4: Comparison of N-PPAC (ours) vs. previous baselines on standard MuJoCo benchmark tasks. **Top**: KL-based methods, **Bottom**: Non-KL-based methods. Both top and bottom plots include N-PPAC (**blue**). BRAC uses the same actor–critic algorithm as N-PPAC, but uses a parametric behavioral policy, and results in slower learning and worse final performance.

sufficient capacity to prevent a collapse in predictive uncertainty away from the training data. Unlike parametric models, whose capacity is limited by their parameterization, a non-parametric model's capacity *increases* with the amount of training data.

Considering a non-parametric behavioral policy, $\pi_0(\cdot \mid \mathbf{s})$ distributed according to a GP:

$$\mathbf{A} \mid \mathbf{s} \sim \pi_0(\cdot \mid \mathbf{s}) = \mathcal{GP}\big(m(\mathbf{s}), k(\mathbf{s}, \mathbf{s}')\big), \tag{5}$$

We can obtain a *non-degenerate* posterior predictive distribution over actions conditioned on the offline data $\mathcal{D}_0 = \{\bar{\mathbf{S}}, \bar{\mathbf{A}}\}$ with actions sampled according to $\mathbf{A} \mid \mathbf{s}, \mathcal{D}_0 \sim \pi_0(\cdot \mid \mathbf{s}, \mathcal{D}_0) = \mathcal{GP}\big(\boldsymbol{\mu}_0(\mathbf{s}), \boldsymbol{\Sigma}_0(\mathbf{s}, \mathbf{s}')\big)$, where

$$\mu(\mathbf{s}) = m(\mathbf{s}) + k(\mathbf{s}, \bar{\mathbf{S}})(k(\bar{\mathbf{S}}, \bar{\mathbf{S}}))^{-1}(\bar{\mathbf{A}} - m(\bar{\mathbf{A}}))$$
$$\Sigma(\mathbf{s}, \mathbf{s}) = k(\mathbf{s}, \mathbf{s}) + k(\mathbf{s}, \bar{\mathbf{S}})(k(\bar{\mathbf{S}}, \bar{\mathbf{S}}))^{-1}k(\bar{\mathbf{S}}, \mathbf{s}).$$

To obtain this posterior distribution, we perform exact Bayesian inference, which naively scales as $\mathcal{O}(N^3)$ in the number of training points N, but Wang et al. [41] show that exact inference in GP regression can be scaled to $N > 1,000,000$. Since expert demonstrations usually contain less than 100k datapoints, non-parametric GP behavioral policies are applicable to a wide array of real-world tasks. For an empirical evaluation of the time complexity of using a GP prior, see Section 5.3

Figure 1 confirms that the non-parametric GP's predictive variance is well-calibrated: It is small near the expert trajectories and large in other parts of the state space. While actor–critic algorithms like SAC implicitly use a uniform prior to explore the state space, using a behavioral policy with well-calibrated predictive variance has the benefit that, close to the offline training data, the online policy learns to match the expert, whereas away from the data, the predictive variance increases and encourages exploration.

Algorithmic Details The use of non-parametric distribution over behavioral policies allows us to formulate a more robust approach to KL-regularized RL, Function-Space Prior Actor–Critic (N-PPAC), which optimizes the KL-augmented RL objective with respect to a non-parametric prior using a standard actor–critic approach with double DQN [11]. Before online training, the online policy is pre-trained to minimize the KL divergence to the behavioral policy on the offline dataset:

$$J_{\mathcal{GP}}(\phi) \doteq \mathbb{E}_{\mathbf{s}\sim\mathcal{D}_0}\left[\mathbb{D}_{\mathrm{KL}}(\pi_\phi(\cdot \mid \mathbf{s}) \parallel \pi_0(\cdot \mid \mathbf{s}))\right]. \tag{6}$$

In contrast to prior work, we make no ad-hoc design choices for online training. Pseudocode is provided in Algorithm 1 (in Appendix I).

5 Experiments

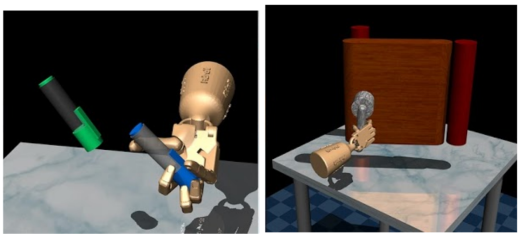
We carry out a comparative empirical evaluation of our proposed approach vis-à-vis related methods that integrate offline data into online training. Appendix H contains a full description of the algorithms

we compare against. We perform experiments on the MuJoCo benchmark suite and the substantially more challenging dexterous hand manipulation suite with sparse rewards. We show that our method can rapidly learn to solve difficult high-dimensional continuous control problems given a only small set of expert demonstrations and that our approach (often significantly) outperforms state-of-the-art methods, including ones that use offline reward information, which our approach does not require. Furthermore, we demonstrate that the behavioral policy's predictive variance is crucial for KL-regularized objectives to learn good policies with expert demonstrations. Finally, we show our method is not hindered by the computational overhead of a GP, and significantly beats standard SAC in terms of wall-clock time and environment samples.

For fair comparison, we use the expert data from Nair et al. [21]. Every experiment uses six seeds and we use a fixed KL-temperature for each environment class. We also present a variant of our approach with automatic temperature tuning with respect to a constraint in Appendix G. The full hyperparameter setup is provided in Appendix J.

5.1 Environments

MuJoCo locomotion tasks. We evaluate our method on three representative tasks: Ant-v2, HalfCheetah-v2, and Walker2d-v2. For each task, we use 15 demonstration trajectories collected by a pre-trained expert, each containing 1000 steps. The results of the empirical evaluation are shown in Figure 4. Our approach consistently outperforms all related methods across all three tasks, successfully accelerating learning from offline data. Notably, we beat methods such as AWAC [21], the previous state-of-the-art, which attempt to eschew the problem of learning behavioral policies but instead use an implicit constraint. Our approach exhibits an increase in stability and higher returns compared to comparable methods such as ABM and BRAC that explicitly regularize the online policy against a parametric behavioral policy and plateau at sub-optimal performance levels as they are being forced to copy poor actions from the behavioral policy away from the expert data. In contrast, using a non-parametric behavioral policy allows us to avoid such undesirable behavior.



(a) pen-binary-v0 **(b)** door-binary-v0

Figure 5: Visualizations of dexterous hand manipulation tasks.

Dexterous hand manipulation tasks. Real-world robot learning is a setting where human demonstration data is readily available and many deep RL approaches fail to learn efficiently. We study this setting in a suite of challenging dexterous manipulation tasks [29] using a 28-DoF five-fingered simulated ADROIT hand. The tasks simulate challenges common to real-world settings with high-dimensional action spaces, complex physics, and a large number of intermittent contact forces.

We consider two tasks in particular: in-hand rotation of a pen to match a target and opening a door by unlatching the handle. We use binary rewards for task completion, which is significantly more challenging than the original setting considered in Rajeswaran et al. [29]. 25 expert demonstrations were provided for each task Rajeswaran et al. [29], each consisting of 200 environment steps which are not fully optimal but do successfully solve the task.

The results of the empirical evaluation are shown in Figure 6. Our approach performs on par or outperforms all related methods on both tasks. Our most notable results are on the door opening task, where we show we can stably achieve 90% success rate in 100k interactions with the environment. For comparison, AWAC required $4\times$ as long to achieve the same level of performance and was far more unstable after its initial offline pre-training phase. The non-KL based approaches fail to learn any meaningful behavior at all.

5.2 Behavioral Policy Variance Ablation

To assess whether the success of non-parametric behavioral policies is due to their predictive variance estimates, as suggested by Proposition 1, or due to better generalization from their predictive means, we perform an ablation study on the predictive variance of the behavioral policy. We *fix the behavioral policy's predictive mean to the mean of the* GP *posterior* and substitute the predictive variances to that of different behavioral policies (parametric and non-parametric). The resultant behavioral policy is

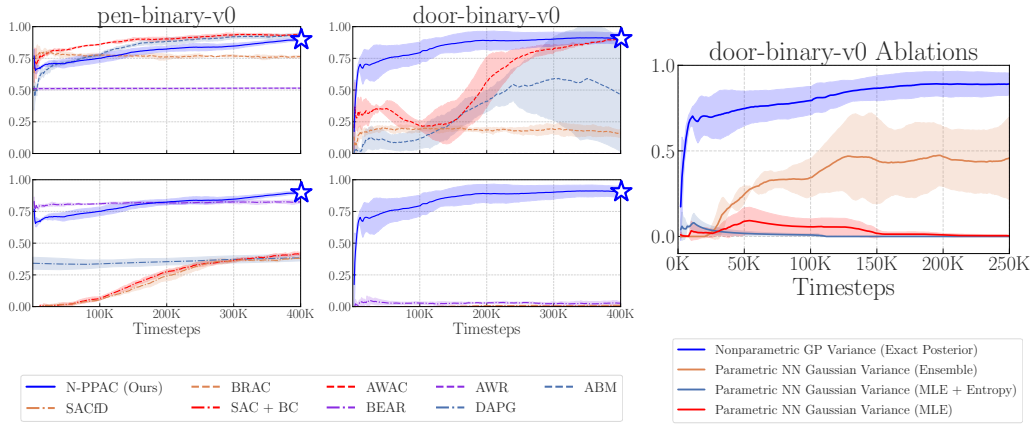


Figure 6: Left & Center: Comparison of N-PPAC (ours) vs. previous baselines on dexterous hand manipulation tasks. **Top**: KL-based methods, **Bottom**: Non-KL-based methods. Both top and bottom plots include N-PPAC (**blue**). **Right**: Comparison of success rates for agents with behavioral policies that have the same GP predictive mean but different predictive variances. Behavioral policies with poorly calibrated predictive variances prevent effective training. Even a probabilistic ensemble falls far short of the GP's performance.

then used in N-PPAC. Figure 6 shows that predictive variances of single and ensembles of parametric models result in online policies that achieve success rates of 0–20% and $\approx 50\%$, respectively, whereas the variance of a non-parametric policy results in a success rate of $> 90\%$. In Appendix D, we show that ensemble policies are marginally better calibrated than parametric neural network policies in that their predictive variance collapses in some but not all regions away from the expert trajectories. These results again demonstrate the importance of accurate predictive uncertainty estimation in allowing the online policy to choose expert actions with high probability and explore away from the data.

5.3 Analysis of Gaussian Process Training Time

We show that only is N-PPAC effective in reducing the number of environment samples required to learn optimal behavior, it is also more efficient in wall-clock time compared to standard SAC. Table 1 presents the number of environment timesteps and wall-clock time needed to reach a baseline level of performance with our method and Soft Actor-Critic (SAC) without expert demonstrations. N-PPAC beats SAC both in terms of sample efficiency and real-time to achieve to same level of performance. We note that SAC completely fails to learn a meaningful policy on the dexterous hand manipulation environments so incorporating expert demonstrations is a crucial for more complex environments.

Table 1: Epochs and wall-block time required to reach certain performance breakpoints by N-PPAC and SAC. N-PPAC beats SAC both in terms of sample efficiency and real-time to achieve to same level of performance.

Environment	Performance Target	SAC		N-PPAC	
		Timesteps	Time (h)	Timesteps	Time (h)
Halfcheetah-v2	7k	1070k	5.9	200k	3.3
Ant-v2	2.5k	1500k	8.3	100k	1.7
Walker2d-v2	4k	1800k	10	200k	3.3
pen-binary-v0	0.7	N/A	inf	50k	0.8
door-binary-v0	0.8	N/A	inf	50k	0.8

6 Conclusion

We identified a previously unrecognized pathology in KL-regularized RL from expert demonstrations and showed that this pathology can significantly impede and even entirely prevent online learning. To remedy the pathology, we proposed the use of non-parametric behavioral policies, which we showed can significantly accelerate and improve online learning and yield online policies that (often significantly) outperform current state-of-the-art methods on challenging continuous control tasks. We hope that this work will encourage further research into the role of model classes in deep reinforcement learning algorithms.

References

[1] Michael Bain and Claude Sammut. A framework for behavioural cloning. In *Machine Intelligence 15*, pages 103–129, 1995.

[2] Abdeslam Boularias, Jens Kober, and Jan Peters. Relative entropy inverse reinforcement learning. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pages 182–189. JMLR Workshop and Conference Proceedings, 2011.

[3] Ivan Bratko, Tanja Urbancic, and Claude Sammut. Behavioural cloning: phenomena, results and problems. *IFAC Proceedings Volumes*, 28(21):143–149, 1995.

[4] Tim Brys, Anna Harutyunyan, Halit Bener Suay, Sonia Chernova, Matthew E Taylor, and Ann Nowé. Reinforcement learning from demonstration through shaping. In *Twenty-fourth international joint conference on artificial intelligence*, 2015.

[5] Thomas Degris, Martha White, and Richard S. Sutton. Off-policy actor-critic. In *Proceedings of the 29th International Coference on International Conference on Machine Learning*, ICML'12, page 179–186, Madison, WI, USA, 2012. Omnipress. ISBN 9781450312851.

[6] Gabriel Dulac-Arnold, Daniel Mankowitz, and Todd Hester. Challenges of real-world reinforcement learning. *arXiv preprint arXiv:1904.12901*, 2019.

[7] Alexandre Galashov, Siddhant M. Jayakumar, Leonard Hasenclever, Dhruva Tirumala, Jonathan Schwarz, Guillaume Desjardins, Wojciech M. Czarnecki, Yee Whye Teh, Razvan Pascanu, and Nicolas Heess. Information asymmetry in kl-regularized RL. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*, 2019.

[8] Yang Gao, Huazhe Xu, Ji Lin, Fisher Yu, Sergey Levine, and Trevor Darrell. Reinforcement learning from imperfect demonstrations. *arXiv preprint arXiv:1802.05313*, 2018.

[9] CW Groetsch. The theory of tikhonov regularization for fredholm equations. *104p, Boston Pitman Publication*, 1984.

[10] Tuomas Haarnoja, Aurick Zhou, Kristian Hartikainen, George Tucker, Sehoon Ha, Jie Tan, Vikash Kumar, Henry Zhu, Abhishek Gupta, Pieter Abbeel, and Sergey Levine. Soft actor-critic algorithms and applications, 2019.

[11] Hado V. Hasselt. Double q-learning. In J. D. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R. S. Zemel, and A. Culotta, editors, *Advances in Neural Information Processing Systems 23*, pages 2613–2621, 2010.

[12] Leslie Pack Kaelbling, Michael L Littman, and Andrew W Moore. Reinforcement learning: A survey. *Journal of artificial intelligence research*, 4:237–285, 1996.

[13] Vijay R Konda and John N Tsitsiklis. Actor-critic algorithms. In *Advances in neural information processing systems*, pages 1008–1014. Citeseer, 2000.

[14] George Konidaris, Scott Kuindersma, Roderic Grupen, and Andrew Barto. Robot learning from demonstration by constructing skill trees. *The International Journal of Robotics Research*, 31 (3):360–375, 2012.

[15] Aviral Kumar, Justin Fu, Matthew Soh, George Tucker, and Sergey Levine. Stabilizing off-policy q-learning via bootstrapping error reduction. In *Advances in Neural Information Processing Systems 32*, pages 11784–11794, 2019.

[16] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 6402–6413, 2017.

[17] Timothy P. Lillicrap, Jonathan J. Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. In *ICLR (Poster)*, 2016.

[18] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin A. Riedmiller. Playing atari with deep reinforcement learning. *CoRR*, abs/1312.5602, 2013.

[19] Kevin P. Murphy. *Machine learning : a probabilistic perspective*. MIT Press, Cambridge, Mass. [u.a.], 2013. ISBN 9780262018029 0262018020.

[20] A. Nair, B. McGrew, M. Andrychowicz, W. Zaremba, and P. Abbeel. Overcoming exploration in reinforcement learning with demonstrations. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6292–6299, 2018.

[21] Ashvin Nair, Murtaza Dalal, Abhishek Gupta, and Sergey Levine. Accelerating online reinforcement learning with offline datasets, 2020.

[22] Nicolás Navarro-Guerrero, Cornelius Weber, Pascal Schroeter, and Stefan Wermter. Real-world reinforcement learning for autonomous humanoid robot docking. *Robotics and Autonomous Systems*, 60(11):1400–1407, 2012.

[23] Andrew Y. Ng and Stuart J. Russell. Algorithms for inverse reinforcement learning. In *Proceedings of the Seventeenth International Conference on Machine Learning*, ICML '00, page 663–670, San Francisco, CA, USA, 2000. Morgan Kaufmann Publishers Inc. ISBN 1558607072.

[24] Aldo Pacchiano, Jack Parker-Holder, Yunhao Tang, Krzysztof Choromanski, Anna Choromanska, and Michael Jordan. Learning to score behaviors for guided policy optimization. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 7445–7454. PMLR, 13–18 Jul 2020.

[25] Xue Bin Peng, Aviral Kumar, Grace Zhang, and Sergey Levine. Advantage-weighted regression: Simple and scalable off-policy reinforcement learning, 2019.

[26] Jan Peters, Sethu Vijayakumar, and Stefan Schaal. Natural actor-critic. In *European Conference on Machine Learning*, pages 280–291. Springer, 2005.

[27] Joaquin Quiñonero Candela and Carl Edward Rasmussen. A unifying view of sparse approximate gaussian process regression. *J. Mach. Learn. Res.*, 6:1939–1959, December 2005. ISSN 1532-4435.

[28] Aravind Rajeswaran, Vikash Kumar, Abhishek Gupta, Giulia Vezzani, John Schulman, Emanuel Todorov, and Sergey Levine. Learning Complex Dexterous Manipulation with Deep Reinforcement Learning and Demonstrations. In *Proceedings of Robotics: Science and Systems (RSS)*, 2018.

[29] Aravind Rajeswaran, Vikash Kumar, Abhishek Gupta, Giulia Vezzani, John Schulman, Emanuel Todorov, and Sergey Levine. Learning complex dexterous manipulation with deep reinforcement learning and demonstrations, 2018.

[30] Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press, 2005. ISBN 026218253X.

[31] Konrad Rawlik, Marc Toussaint, and Sethu Vijayakumar. On stochastic optimal control and reinforcement learning by approximate inference. *Proceedings of Robotics: Science and Systems VIII*, 2012.

[32] Michael T Rosenstein, Andrew G Barto, Jennie Si, Andy Barto, and Warren Powell. Supervised actor-critic reinforcement learning. *Learning and Approximate Dynamic Programming: Scaling Up to the Real World*, pages 359–380, 2004.

[33] Stefan Schaal et al. Learning from demonstration. *Advances in neural information processing systems*, pages 1040–1046, 1997.

[34] John Schulman, Xi Chen, and Pieter Abbeel. Equivalence between policy gradients and soft q-learning. *arXiv preprint arXiv:1704.06440*, 2017.

[35] Noah Siegel, Jost Tobias Springenberg, Felix Berkenkamp, Abbas Abdolmaleki, Michael Neunert, Thomas Lampe, Roland Hafner, Nicolas Heess, and Martin Riedmiller. Keep doing what worked: Behavior modelling priors for offline reinforcement learning. In *International Conference on Learning Representations*, 2020.

[36] Alex Smola, Arthur Gretton, Le Song, and Bernhard Schölkopf. A hilbert space embedding for distributions. In Marcus Hutter, Rocco A. Servedio, and Eiji Takimoto, editors, *Algorithmic Learning Theory*, pages 13–31, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg. ISBN 978-3-540-75225-7.

[37] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. The MIT Press, second edition, 2018.

[38] Gerald Tesauro. Temporal difference learning and td-gammon. *Communications of the ACM*, 38(3):58–68, 1995.

[39] Emanuel Todorov. Linearly-solvable markov decision problems. In B. Schölkopf, J. Platt, and T. Hoffman, editors, *Advances in Neural Information Processing Systems*, volume 19, pages 1369–1376. MIT Press, 2007.

[40] Joost van Amersfoort, Lewis Smith, Andrew Jesson, Oscar Key, and Yarin Gal. Improving deterministic uncertainty estimation in deep learning for classification and regression, 2021.

[41] Ke Wang, Geoff Pleiss, Jacob Gardner, Stephen Tyree, Kilian Q Weinberger, and Andrew Gordon Wilson. Exact gaussian processes on a million data points. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32, 2019.

[42] Yifan Wu, George Tucker, and Ofir Nachum. Behavior regularized offline reinforcement learning. *arXiv preprint arXiv:1911.11361*, 2019.

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] See Section 4.1.
   (c) Did you discuss any potential negative societal impacts of your work? [N/A] It's a general-purpose algorithm.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes] See Proposition 1.
   (b) Did you include complete proofs of all theoretical results? [Yes] See Appendix A.
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] In supplementary material.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See Appendix J.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [No] We did not need significant amounts of compute.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...

(a) If your work uses existing assets, did you cite the creators? [Yes] Cited Rajeswaran et al. and Nair et al. for the datasets.

(b) Did you mention the license of the assets? [Yes] The offline datasets are provided in a repository under an Apache License 2.0. This is noted in Appendix J.

(c) Did you include any new assets either in the supplemental material or as a URL? [No]

(d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]

(e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]

5. If you used crowdsourcing or conducted research with human subjects...

(a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]

(b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]

(c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]