ON PATHOLOGIES IN KL-REGULARIZED REINFORCEMENT LEARNING FROM EXPERT DEMONSTRATIONS

Tim G. J. Rudner[*†] **Cong Lu**[*] **Michael A. Osborne** **Yarin Gal** **Yee Whye Teh**

University of Oxford, Oxford, UK

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 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 (Tesauro, 1995; Kaelbling et al., 1996; Mnih et al., 2013; Sutton & Barto, 2018) 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 (Navarro-Guerrero et al., 2012; Dulac-Arnold et al., 2019). To improve the robustness and sample efficiency of modern reinforcement learning algorithms, a significant body of prior work has explored approaches for efficiently incorporating expert demonstrations into the learning process (Schaal et al., 1997; Konidaris et al., 2012; Brys et al., 2015; Gao et al., 2018).

KL-regularized RL is a particularly successful approach for doing so (Ng & Russell, 2000; Todorov, 2007; Boularias et al., 2011). 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 (Wu et al., 2019), ABM (Siegel et al., 2020), and AWAC (Nair et al., 2020), 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.

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 show that behavioral policy classes commonly used in KL-regularized RL exhibit poorly calibrated predictive uncertainty estimates and experience a collapse in predictive variance about states away from the expert demonstrations. We demonstrate both theoretically and empirically that when this occurs, KL-regularized RL algorithms suffer from pathological behavior in online training. Finally, we show that the pathology can be remedied by specifying a (tractable) *non-parametric* behavioral policy whose predictive variance is guaranteed not to collapse about states

[*]Equal contribution.

[†]Corresponding author: tim.rudner@cs.ox.ac.uk

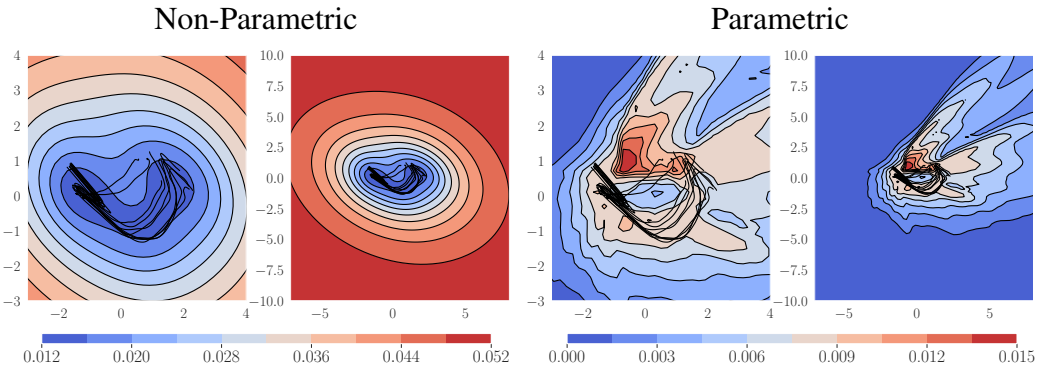


Figure 1: Predictive variances of non-parametric and parametric behavioral policies on a low dimensional representation of a 39-dimensional dexterous hand manipulation state space (see "door-binary-v0" in Appendix G). **Left**: Non-parametric Gaussian process posterior behavioral policy $\pi_{\mathcal{GP}}(\cdot \mid \mathbf{s}, \mathcal{D}_0) = \mathcal{GP}(\boldsymbol{\mu}_0(\mathbf{s}), \boldsymbol{\Sigma}_0(\mathbf{s}, \mathbf{s}'))$. **Right**: Parametric neural network Gaussian behavioral policy $\pi_\psi(\cdot \mid \mathbf{s}) = \mathcal{N}(\boldsymbol{\mu}_\psi(\mathbf{s}), \boldsymbol{\sigma}_\psi(\mathbf{s}))$. 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.

off the expert trajectory, and that doing so results in online policies that significantly outperform state-of-the-art approaches on a range of challenging locomotion and dexterous manipulation tasks *without* ad-hoc design choices. In this way, we present 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 Identifying Pathological Behavior in KL-Regularized RL

In this section, we investigate the effect of KL-regularization on RL 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 show that behavioral policies with small predictive variance estimates cause the KL-regularization term and, as a result, the gradients of the policy objective to blow up. We confirm this failure empirically in Figures 1 and 4 and demonstrate that it results in slow, unstable, and suboptimal online training in Figures 2 and 3.

Notation. We consider the standard reinforcement learning setting in which an agent interacts with a discounted Markov Decision Process (MDP) (Sutton & Barto, 2018) given by the 5-tuple $(\mathcal{S}, \mathcal{A}, p, r, \gamma)$, where $\mathcal{S}$ and $\mathcal{A}$ are the state and action spaces, $p(\cdot \mid \mathbf{s}_t, \mathbf{a}_t)$ are the transition dynamics, $r(\mathbf{s}_t, \mathbf{a}_t)$ is the reward function, and γ is a discount factor. 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 RL 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, $\rho_\pi(\cdot)$.

KL-Regularized Objectives in Reinforcement Learning. We consider settings where we have a set of expert demonstrations *without reward*, $\mathcal{D}_0 = \{(\mathbf{s}_n, \mathbf{a}_n)\}_{n=1}^N = \{\bar{\mathbf{S}}, \bar{\mathbf{A}}\}$, from which we can learn a policy via behavioral cloning, $\pi_0 : \mathcal{S} \to \mathcal{A}$, that maps the states in the expert demonstrations to their corresponding actions (Bain & Sammut, 1995; Bratko et al., 1995).

KL-regularized reinforcement learning (Todorov, 2007; Rawlik et al., 2012; Schulman et al., 2017; Galashov et al., 2019) allows a learner to build on a cloned behavior by interacting with the environment. Given a reference policy π_0 and temperature parameter α, 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. The resulting discounted return from time step $t \in \mathbb{N}_0$ is then given by $\tilde{R}(\boldsymbol{\tau}_t) = \sum_{k=t}^{\infty} \gamma^k \big[r(\mathbf{s}_k, \mathbf{a}_k) - \alpha \mathbb{D}_{\mathrm{KL}}(\pi(\cdot \mid \mathbf{s}_k) \parallel \pi_0(\cdot \mid \mathbf{s}_k))\big]$ and the reinforcement learning objective becomes $\tilde{J}_\pi(\boldsymbol{\tau}_0) = \mathbb{E}_{\rho_\pi(\boldsymbol{\tau}_0)}[\tilde{R}(\boldsymbol{\tau}_0)]$. We may optimize this objective via actor–critic algorithms, which alternate between policy improvement and policy evaluation steps (see Appendix A.1 for further details).

When are KL-Regularized RL Objectives Meaningful? We start by considering key properties of the KL divergence which can lead to potential failure modes in KL-regularized objectives.

Definition 1 (Kullback-Leibler Divergence, Gray (2011))**.** *Let Q and P be probability measures over a measurable space $(\mathcal{X}, \mathcal{F})$ and $\frac{dQ}{dP}$ the Radon-Nikodym derivative of Q with respect to P. Then the Kullback-Leibler divergence from Q to P is defined as $\mathbb{D}_{\mathrm{KL}}(Q \,||\, P) \stackrel{\text{def}}{=} \int_{\mathcal{X}} \log \frac{dQ}{dP}\, dQ < \infty$, if Q is absolutely continuous with respect to P, and infinite otherwise.*

As this definition shows, the KL divergence is infinite if the probability measure Q is not absolutely continuous with respect to the probability measure P. For KL-regularized RL, this means that in order for the learning objective to be meaningful (that is, non-infinite), the online policy must place zero measure on any set on which the behavioral policy places zero measure. While Gaussian behavioral and online policies, often chosen in practice, are absolutely continuous with respect to one another, a Gaussian online policy $\pi_\phi(\cdot \,|\, \mathbf{s}_t)$ loses absolute continuity with respect to $\pi_0(\cdot \,|\, \mathbf{s}_t)$ as the predictive variance $\boldsymbol{\sigma}_0^2$ of $\pi_0(\cdot \,|\, \mathbf{s}_t)$ tends to zero, and so $\mathbb{D}_{\mathrm{KL}}(\pi_\phi(\cdot \,|\, \mathbf{s}_t) \parallel \pi_0(\cdot \,|\, \mathbf{s}_t)) \to \infty$ as $\boldsymbol{\sigma}_0^2 \to 0$. This naturally raises the question how very small $\boldsymbol{\sigma}_0^2$ may affect online training.

Exploding Gradients in KL-Regularized RL. To understand how small predictive variances in behavioral policies affect online training in KL-regularized RL, we consider the contribution of the behavioral policy's variance to the gradients of the policy objective (given Equation (A.5)):

Proposition 1 (Exploding Gradients in KL-Regularized RL)**.** *Let $\pi_0(\cdot \,|\, \mathbf{s})$ be a Gaussian behavioral policy with mean $\boldsymbol{\mu}_0$ and variance $\boldsymbol{\sigma}_0^2$, and let $\pi_\phi(\cdot \,|\, \mathbf{s})$ 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*

$$\begin{aligned}\hat{\nabla}_\phi J_\pi(\phi) = \big(&\alpha \nabla_{\mathbf{a}_t} \log \pi_\phi(\mathbf{a}_t \mid \mathbf{s}_t) - \alpha \nabla_{\mathbf{a}_t} \log \pi_0(\mathbf{a}_t \mid \mathbf{s}_t) \\ &- \nabla_{\mathbf{a}_t} Q(\mathbf{s}_t, \mathbf{a}_t)\big) \nabla_\phi f_\phi(\epsilon_t; \mathbf{s}_t) + \alpha \nabla_\phi \log \pi_\phi(\mathbf{a}_t \mid \mathbf{s}_t)\end{aligned} \tag{1}$$

with $\nabla_{\mathbf{a}_t} \log \pi_0(\mathbf{a}_t \,|\, \mathbf{s}_t) = -\frac{\mathbf{a}_t - \boldsymbol{\mu}_0}{\boldsymbol{\sigma}_0^2}$. 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,

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

Proof. See Appendix B.1 for a proof and Appendix B.2 for empirical confirmation of the result. □

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.

Empirical Confirmation of Collapsing Variance. The most commonly used method for training behaviorally cloned stochastic policies is maximum likelihood estimation (MLE) (Wu et al., 2019; Siegel et al., 2020). Figure 1 demonstrates the collapse in predictive variance of a neural network trained via MLE in a low-dimensional representation of the "door-binary-v0" environment. It shows that while the predictive variance is small close to the expert trajectories (shown as black lines), it rapidly decreases further away from them. A discussion on why this occurs is included in Appendix C, and confirmation of variance collapse in other environments is presented in Appendix E.

3 Fixing the Pathology

Non-Parametric Behavioral Policies To fix the pathology identified in Proposition 1 and stabilize KL-regularized RL, we propose N-PPAC: Non-Parametric Prior Actor–Critic. To avoid a collapse in behavioral policies' predictive variance estimates, we consider a non-parametric Gaussian process (GP) behavioral policy. GPs are distributions over functions defined by a mean and covariance function. If a GP's covariance function is non-parametric, that is, constructed from infinitely many basis functions, the resulting GP is considered non-degenerate. Unlike parametric models, whose capacity is limited by their parameterization, the capacity of a non-degenerate GPs *increases* with the amount of training data, preventing a collapse in predictive uncertainty away from the training data. Figure 1 and Figure 8 in Appendix E confirm that a non-parametric behavioral policy's predictive variance is indeed well-calibrated: It is small near the expert trajectories and large in other parts of the state space. The analytical expression of the GP behavioral policy and pseudocode for N-PPAC are provided in Appendix F.

Comparative Empirical Evaluation 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

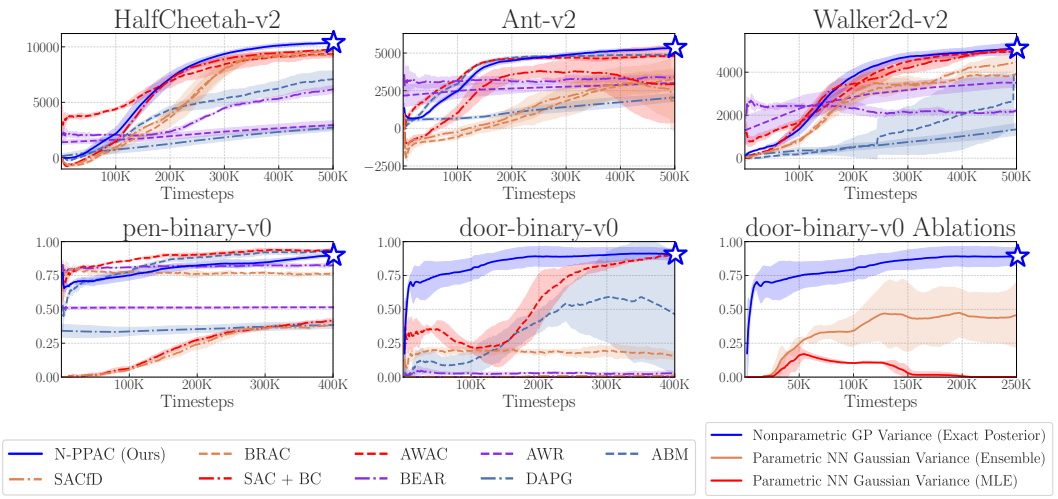


Figure 2: Top & Bottom Left: Comparison of N-PPAC (ours) vs. previous baselines on standard MuJoCo benchmark tasks and dexterous hand manipulation tasks across six seeds. **Bottom Right**: Comparison of success rates for agents with behavioral policies that have the same GP predictive mean but different predictive variances. Behavioral policies with poorly calibrated predictive variances prevent effective training.

contains a full description of the algorithms we compare against. We perform experiments on the MuJoCo benchmark suite and on the substantially more challenging dexterous hand manipulation suite with sparse rewards. For fair comparison, we use the expert data from Nair et al. (2020). The full environment setup is given in Appendix G and the hyperparameter setup is provided in Appendix I.

The results of the empirical evaluation are shown in Figure 2. Our approach consistently outperforms all related methods across all five tasks, successfully accelerating learning from expert demonstrations. Compared to AWAC (Nair et al., 2020), the previous state-of-the-art, we do not require additional generation of data from the expert demonstration set or any offline reward data. Our approach significantly improves on 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. Our most notable results are on the door opening task, where we show we can stably achieve a 90% success rate within 100k interactions with the environment, which is almost $4\times$ faster than AWAC.

Behavioral Policy Variance Ablations Finally, we demonstrate that the behavioral policy's predictive variance is in fact crucial for KL-regularized objectives to learn good policies with expert demonstrations, as suggested by Proposition 1. We perform an ablation study in Figure 2 where we fix the behavioral policy's predictive mean to the mean of the GP posterior and train an online policy using different behavioral policy predictive variances (parametric and non-parametric). The 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 only 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 far away from the data.

4 Conclusion

We identified a hitherto unrecognized pathology in KL-regularized RL from expert demonstrations and demonstrated that this pathology can significantly impede and even entirely prevent online learning, as shown in Figures 2 and 3. 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.

ACKNOWLEDGMENTS

We are grateful to Ashvin Nair for sharing his code and results as well as for sharing helpful insights about the dexterous hand manipulation suite. We also thank Clare Lyle, Charline Le Lan, and Angelos Filos for detailed feedback on an early draft of this paper, Avi Singh for early discussions about behavioral cloning in entropy-regularized RL, and Tim Pearce for a useful discussion on the role of good models in RL. Tim G. J. Rudner is funded by the Rhodes Trust and the Engineering and Physical Sciences Research Council (EPSRC). Cong Lu is funded by the Engineering and Physical Sciences Research Council (EPSRC). We gratefully acknowledge donations of computing resources by the Alan Turing Institute.

REFERENCES

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

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

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

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.

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, pp. 179–186, Madison, WI, USA, 2012. Omnipress. ISBN 9781450312851.

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

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.

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

Robert M. Gray. *Entropy and Information Theory*. Springer Publishing Company, Incorporated, 2nd edition, 2011. ISBN 9781441979698.

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

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.

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

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

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

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.

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*, pp. 11784–11794, 2019.

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 (eds.), *Advances in Neural Information Processing Systems 30*, pp. 6402–6413, 2017.

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.

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.

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)*, pp. 6292–6299, 2018.

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

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.

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

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 (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 7445–7454. PMLR, 13–18 Jul 2020.

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

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

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.

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)*, 2018a.

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, 2018b.

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

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.

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*, pp. 359–380, 2004.

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

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

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.

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

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

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

Emanuel Todorov. Linearly-solvable markov decision problems. In B. Schölkopf, J. Platt, and T. Hoffman (eds.), *Advances in Neural Information Processing Systems*, volume 19, pp. 1369–1376. MIT Press, 2007.

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 (eds.), *Advances in Neural Information Processing Systems*, volume 32, 2019.

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

SUPPLEMENTARY MATERIAL

A KL-REGULARIZED OBJECTIVES IN REINFORCEMENT LEARNING

Given a reference policy π_0 and temperature parameter α, 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. The resulting discounted return from time step $t \in \mathbb{N}_0$ is then given by

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

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) (Haarnoja et al., 2019) 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.

A.1 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 (Konda & Tsitsiklis, 2000; Rosenstein et al., 2004; Peters et al., 2005; Degris et al., 2012) 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

$$V^\pi(\mathbf{s}_t) \stackrel{\text{def}}{=} \mathbb{E}_{\mathbf{a}_t \sim \pi}[Q^\pi(\mathbf{s}_t, \mathbf{a}_t)] - \alpha \mathbb{D}_{\mathrm{KL}}\big(\pi(\cdot \mid \mathbf{s}_t) \,||\, \pi_0(\cdot \,|\, \mathbf{s}_t)\big) \tag{A.2}$$

with a recursively defined Q-function

$$Q^\pi(\mathbf{s}_t, \mathbf{a}_t) \stackrel{\text{def}}{=} r(\mathbf{s}_t, \mathbf{a}_t) + \gamma \, \mathbb{E}_{\mathbf{s}_{t+1} \sim p}[V^\pi(\mathbf{s}_{t+1})]. \tag{A.3}$$

Instead of directly optimizing the objective function $\tilde{J}_\pi$ via the policy gradient, actor–critic methods alternate between policy evaluation and policy improvement (Degris et al., 2012; Haarnoja et al., 2019):

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

$$J_Q(\theta) \stackrel{\text{def}}{=} \mathbb{E}_{(\mathbf{s}_t, \mathbf{a}_t) \sim \mathcal{D}} \Big[\big(Q_\theta(\mathbf{s}_t, \mathbf{a}_t) - (r(\mathbf{s}_t, \mathbf{a}_t) + \gamma \mathbb{E}_{\mathbf{s}_{t+1} \sim p}[V_{\bar{\theta}}(\mathbf{s}_{t+1})]) \big)^2 \Big], \tag{A.4}$$

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,

$$J_\pi(\phi) \stackrel{\text{def}}{=} \mathbb{E}_{\mathbf{s}_t \sim \mathcal{D}} \left[\alpha \mathbb{D}_{\mathrm{KL}}(\pi_\phi(\cdot \,|\, \mathbf{s}_t) \parallel \pi_0(\cdot \,|\, \mathbf{s}_t)) \right] - \mathbb{E}_{\mathbf{s}_t \sim \mathcal{D}} \left[\mathbb{E}_{\mathbf{a}_t \sim \pi_\phi} \left[Q_\theta(\mathbf{s}_t, \mathbf{a}_t) \right] \right], \tag{A.5}$$

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.

B Proof and Empirical Confirmation of Proposition 1

B.1 Proof

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*

$$\begin{aligned}\hat{\nabla}_\phi J_\pi(\phi) = \big(&\alpha \nabla_{\mathbf{a}_t} \log \pi_\phi(\mathbf{a}_t \mid \mathbf{s}_t) - \alpha \nabla_{\mathbf{a}_t} \log \pi_0(\mathbf{a}_t \mid \mathbf{s}_t) \\ &- \nabla_{\mathbf{a}_t} Q(\mathbf{s}_t, \mathbf{a}_t)\big) \nabla_\phi f_\phi(\epsilon_t; \mathbf{s}_t) + \alpha \nabla_\phi \log \pi_\phi(\mathbf{a}_t \mid \mathbf{s}_t)\end{aligned} \tag{B.6}$$

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,

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

Proof. The policy loss, as given in Equation (A.5), is:

$$J_\pi(\phi) = \mathbb{E}_{\mathbf{s}_t \sim D}\left[\mathbb{D}_{\text{KL}}\big(\pi_\phi(\cdot \mid \mathbf{s}_t) \,||\, \pi_0(\cdot \mid \mathbf{s}_t)\big)\right] - \mathbb{E}_{\mathbf{s}_t \sim D}\left[\mathbb{E}_{\mathbf{a}_t \sim \pi_\phi}\left[Q_\theta(\mathbf{s}_t, \mathbf{a}_t)\right]\right]. \tag{B.7}$$

To obtain a lower-variance gradient estimator, the policy is reparameterized using a neural network transformation

$$\mathbf{a}_t = f_\phi(\epsilon_t; \mathbf{s}_t) \tag{B.8}$$

where ϵ_t is an input noise vector. Following Haarnoja et al. (2019), we can now rewrite Equation (B.7) as

$$J_\pi(\phi) = \mathbb{E}_{\mathbf{s}_t \sim D, \epsilon_t \sim \mathcal{N}}\left[\alpha\big(\log \pi_\phi(f_\phi(\epsilon_t; \mathbf{s}_t) \mid \mathbf{s}_t) - \log \pi_0(f_\phi(\epsilon_t; \mathbf{s}_t) \mid \mathbf{s}_t)\big) - Q(\mathbf{s}_t, f_\phi(\epsilon_t; \mathbf{s}_t))\right] \tag{B.9}$$

where D is a replay buffer and π_ϕ is defined implicitly in terms of f_ϕ. We can approximate the gradient of Equation (B.9) with

$$\begin{aligned}\hat{\nabla}_\phi J_\pi(\phi) = \big(&\alpha \nabla_{\mathbf{a}_t} \log \pi_\phi(\mathbf{a}_t \mid \mathbf{s}_t) - \alpha \nabla_{\mathbf{a}_t} \log \pi_0(\mathbf{a}_t \mid \mathbf{s}_t) \\ &- \nabla_{\mathbf{a}_t} Q(\mathbf{s}_t, \mathbf{a}_t)\big) \nabla_\phi f_\phi(\epsilon_t; \mathbf{s}_t) + \alpha \nabla_\phi \log \pi_\phi(\mathbf{a}_t \mid \mathbf{s}_t)\end{aligned} \tag{B.10}$$

Next, consider the term $\nabla_{\mathbf{a}_t} \log \pi_0(\mathbf{a}_t \mid \mathbf{s}_t)$ for a Gaussian policy:

$$\log \pi_0(\mathbf{a}_t \mid \mathbf{s}_t) = \log\left(\frac{1}{\boldsymbol{\sigma}_0 \sqrt{2\pi}}\right) - \frac{1}{2}\left(\frac{\mathbf{a}_t - \boldsymbol{\mu}_0}{\boldsymbol{\sigma}_0}\right)^2 \tag{B.11}$$

Thus,

$$\nabla_{\mathbf{a}_t} \log \pi_0(\mathbf{a}_t \mid \mathbf{s}_t) = -\frac{\mathbf{a}_t - \boldsymbol{\mu}_0}{\boldsymbol{\sigma}_0^2}. \tag{B.12}$$

For fixed $|\mathbf{a}_t - \boldsymbol{\mu}_0|$, $\nabla_{\mathbf{a}_t} \log(\pi_0(\mathbf{a}_t \mid \mathbf{s}_t))$ grows as $\mathcal{O}(\boldsymbol{\sigma}_0^{-2})$, and so,

$$|\hat{\nabla}_\phi J_\pi(\phi)| \to \infty \quad \text{as} \quad \boldsymbol{\sigma}_0^2 \to 0. \tag{B.13}$$

when $\nabla_\phi f_\phi(\epsilon_t; \mathbf{s}_t) \neq 0$. □

B.2 Empirical Confirmation

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 $\{1 \times 10^{-2}, 5 \times 10^{-3}, 1 \times 10^{-3}\}$ (following a similar implementation in Nair et al. (2020)). [1] The results are shown in Figure 3, which

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

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.

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. Figure 3 also shows that an analogous relationship exists for the gradients of the Q-function loss.

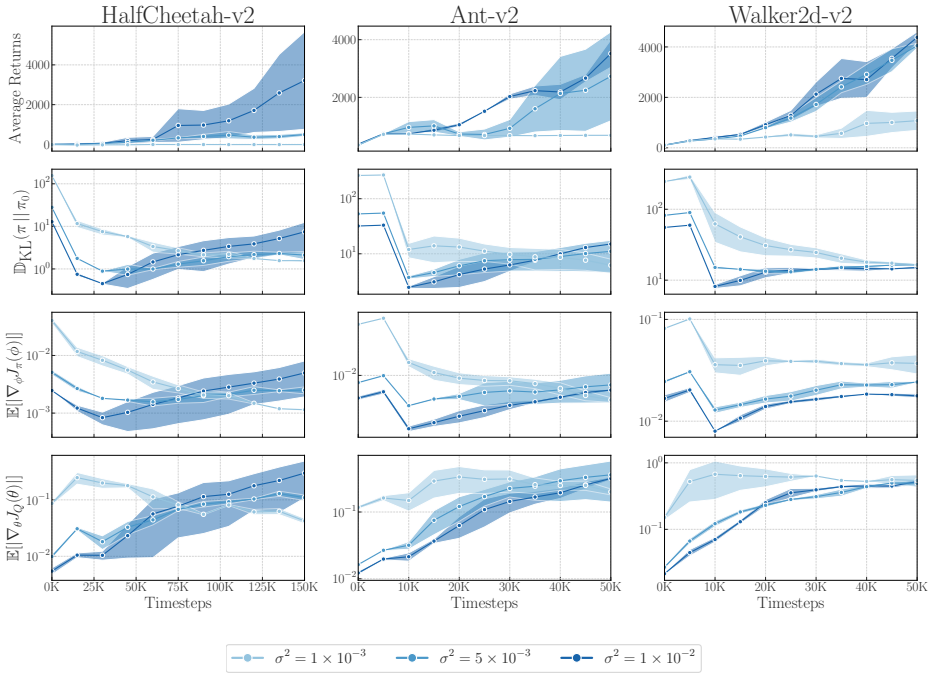


Figure 3: Ablation study showing the effect of predictive variance collapse on the performance of KL-regularized RL on MuJoCo benchmarks. Policies shown from dark to light in order of decreasing constant predictive variance, simulating training under maximum likelihood estimation. The plots show the average return of the learned policy, magnitude of the KL penalty, and magnitude of the policy and Q-function gradients during online training.

C Training Parametric Behavioral Policies

C.1 Maximum Likelihood Estimation Without Regularization

The most commonly used method for training behaviorally cloned stochastic policies is maximum likelihood estimation (Wu et al., 2019; Siegel et al., 2020), where we seek $\pi_0 \stackrel{\text{def}}{=} \pi_{\psi^\star}$ with $\psi^\star \stackrel{\text{def}}{=} \arg\max_\psi \left\{\mathbb{E}_{(\mathbf{s},\mathbf{a})\sim\mathcal{D}_0}[\log \pi_\psi(\mathbf{a}\,|\,\mathbf{s})]\right\}$. 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 its mean and variance 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

$$-\log \pi_\psi(\bar{\mathbf{A}} \mid \bar{\mathbf{S}}) = \sum_{n=1}^{N} \frac{\log \boldsymbol{\sigma}_\psi^2(\mathbf{s}_n)}{2} + \frac{(\mathbf{a}_n - \boldsymbol{\mu}_\psi(\mathbf{s}_n))^2}{2\boldsymbol{\sigma}_\psi^2(\mathbf{s}_n)} + C, \tag{C.14}$$

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 (Quiñonero Candela & Rasmussen, 2005).

Figure 4 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.

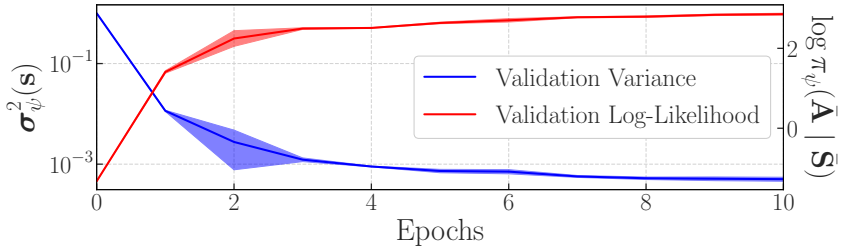


Figure 4: Collapse in the predictive variance (in **blue**) of a Gaussian behavioral policy parameterized by a neural network when training via maximum likelihood estimation. Lines and shaded regions denote means and standard deviations over five random seeds, respectively.

C.2 Maximum Likelihood Estimation With Regularization

To prevent a collapse in the behavioral policy's predictive variance, prior work proposed adding entropy or Tikhonov regularization to the MLE objective (Wu et al., 2019). However, doing so does not achieve a collapse in predictive variance off the expert demonstration trajectories, as we show in Appendix C.

To address the collapse in predictive variance away from the offline dataset under MLE training seen in Figure 1, Wu et al. (2019) in practice augment the usual MLE loss with an entropy bonus as follows:

$$\pi_0 \stackrel{\text{def}}{=} \pi_{\psi^\star} \quad \text{with} \quad \psi^\star \stackrel{\text{def}}{=} \arg\max_{\psi} \left\{ \mathbb{E}_{(\mathbf{s},\mathbf{a})\sim\mathcal{D}}[\log \pi_\psi(\mathbf{a}\,|\,\mathbf{s}) + \beta \mathcal{H}(\pi_\psi(\cdot\,|\,\mathbf{s}))] \right\}. \tag{C.15}$$

where β is temperature tuned to a entropy constraint similar to Haarnoja et al. (2019). The entropy bonus is estimated by sampling from the behavioral policy as

$$\mathcal{H}(\pi_\psi(\cdot\,|\,\mathbf{s})) = \mathbb{E}_{\mathbf{a}_\psi\sim\pi_\psi}[-\log \pi_\psi(\mathbf{a}_\psi\,|\,\mathbf{s})] \tag{C.16}$$

Figure 5 shows the predictive variances of behavioral policies trained on expert demonstrations for the "door-binary-v0" environment with various entropy coefficients β. Whilst entropy regularization partially mitigates the collapse of predictive variance away from the expert demonstrations, we still observe the wrong trend similar to Figure 1 with predictive variances high near the expert demonstrations and low on unseen data. The variance surface also becomes more poorly behaved with "islands" of high predictive variance appearing away from the data.

We may also add Tikhonov regularization (Groetsch, 1984) to the MLE objective, explicitly,

$$\pi_0 \stackrel{\text{def}}{=} \pi_{\psi^\star} \quad \text{with} \quad \psi^\star \stackrel{\text{def}}{=} \arg\max_{\psi} \left\{ \mathbb{E}_{(\mathbf{s},\mathbf{a})\sim\mathcal{D}}[\log \pi_\psi(\mathbf{a}\,|\,\mathbf{s}) - \lambda \psi^T \psi] \right\}. \tag{C.17}$$

where λ is the regularization coefficient.

Figure 6 shows the predictive variances of behavioral policies trained on expert demonstrations for the "door-binary-v0" environment with varying Tikhonov regularization coefficients λ. Similarly, Tikhonov regularization does not resolve the issue with calibration of uncertainties. We also observe that too high a regularization strength causes the model to underfit to the variances of the data.

C.3 Probabilistic Ensembles

A widely used method to obtain predictive uncertainty estimates from neural networks in regression settings are probabilistic ensembles (Lakshminarayanan et al., 2017), 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-calibrates uncertainty estimates.

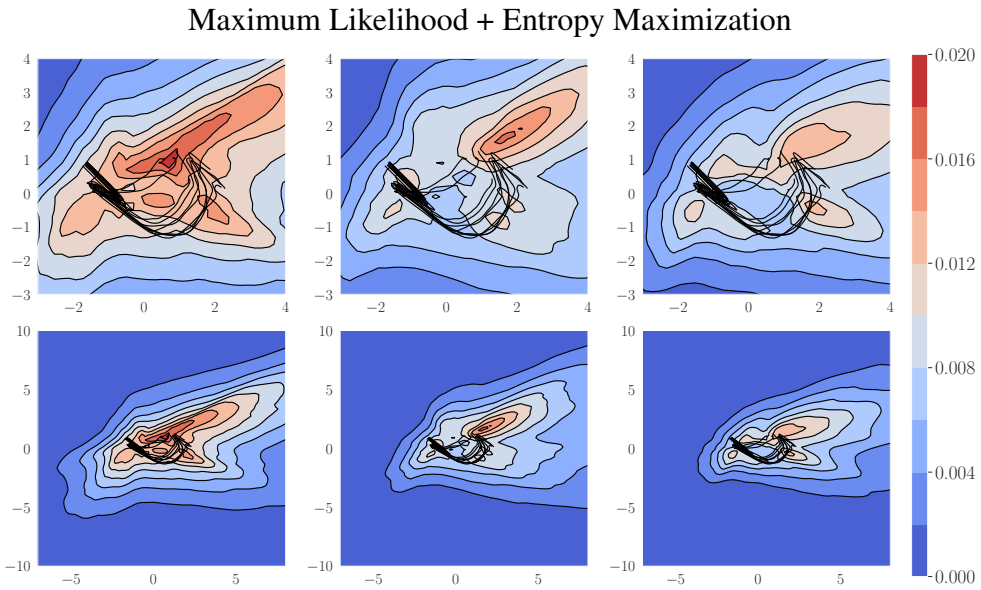
In our experiments, we consider an ensemble of parametric neural network Gaussian policies

$$\pi_{\psi^{1:K}}(\cdot\,|\,\mathbf{s}) \stackrel{\text{def}}{=} \mathcal{N}(\boldsymbol{\mu}_{\psi^{1:K}}(\mathbf{s}), \boldsymbol{\sigma}^2_{\psi^{1:K}}(\mathbf{s})) \tag{C.18}$$

with

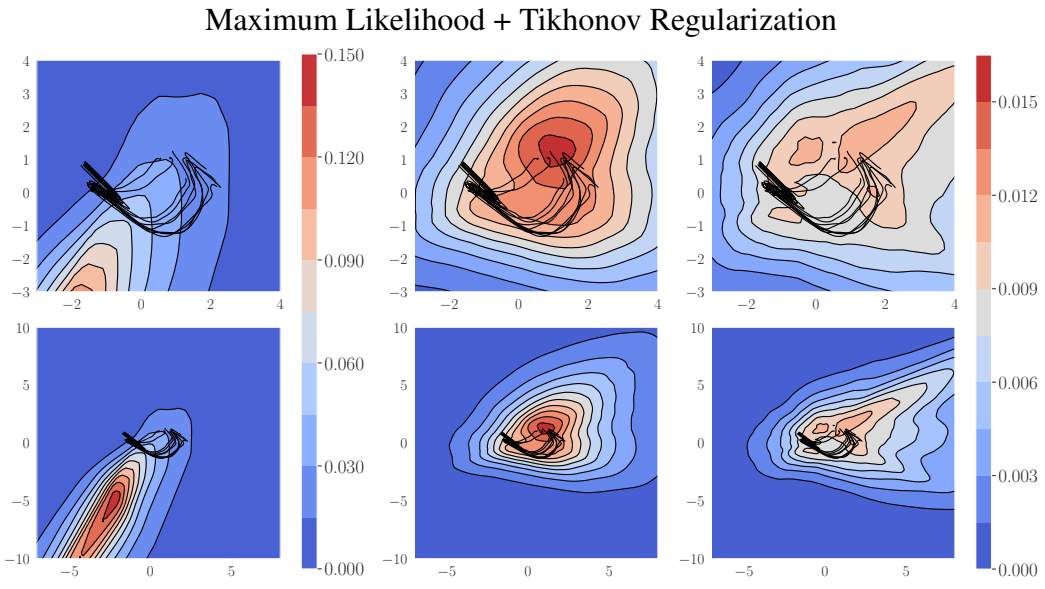
$$\mu_{\psi^{1:K}}(\mathbf{s}) \stackrel{\text{def}}{=} \frac{1}{K}\sum_{k=1}^{K} \boldsymbol{\mu}_{\psi^k}(\mathbf{s}) \tag{C.19}$$

$$\boldsymbol{\sigma}^2_{\psi^{1:K}}(\mathbf{s}) \stackrel{\text{def}}{=} \frac{1}{K}\sum_{k=1}^{K} \left(\boldsymbol{\sigma}^2_{\psi^k}(\mathbf{s}) + \boldsymbol{\mu}^2_{\psi^k}(\mathbf{s}) \right) - \mu^2_{\psi^{1:K}}(\mathbf{s}) \tag{C.20}$$



(a) $\beta = 10^{-2}$ **(b)** $\beta = 10^{-3}$ **(c)** $\beta = 10^{-4}$

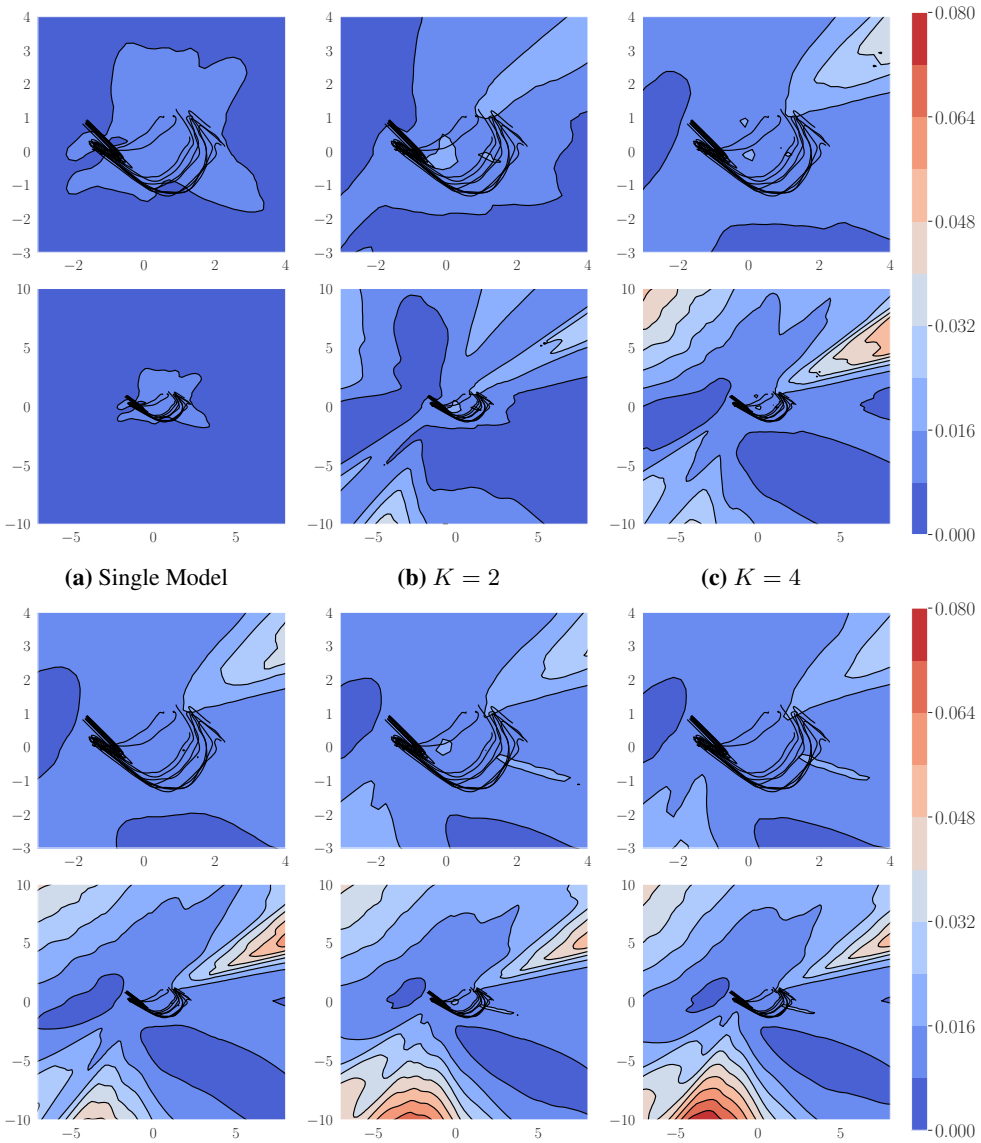
Figure 5: Predictive variances of parametric neural network Gaussian behavioral policies $\pi_\psi(\cdot \,|\, \mathbf{s}) = \mathcal{N}(\boldsymbol{\mu}_\psi(\mathbf{s}), \boldsymbol{\sigma}_\psi(\mathbf{s}))$ on a low-dimensional representation of the "door-binary-v0" environment, trained with different entropy regularization coefficients β.



(a) $\lambda = 10^{-2}$ **(b)** $\lambda = 10^{-3}$ **(c)** $\lambda = 10^{-4}$

Figure 6: Predictive variances of parametric neural network Gaussian behavioral policies $\pi_\psi(\cdot \,|\, \mathbf{s}) = \mathcal{N}(\boldsymbol{\mu}_\psi(\mathbf{s}), \boldsymbol{\sigma}_\psi(\mathbf{s}))$ on a low-dimensional representation of the "door-binary-v0" environment, trained with different Tikhonov regularization coefficients λ.

D ENSEMBLE PARAMETRIC BEHAVIORAL POLICIES



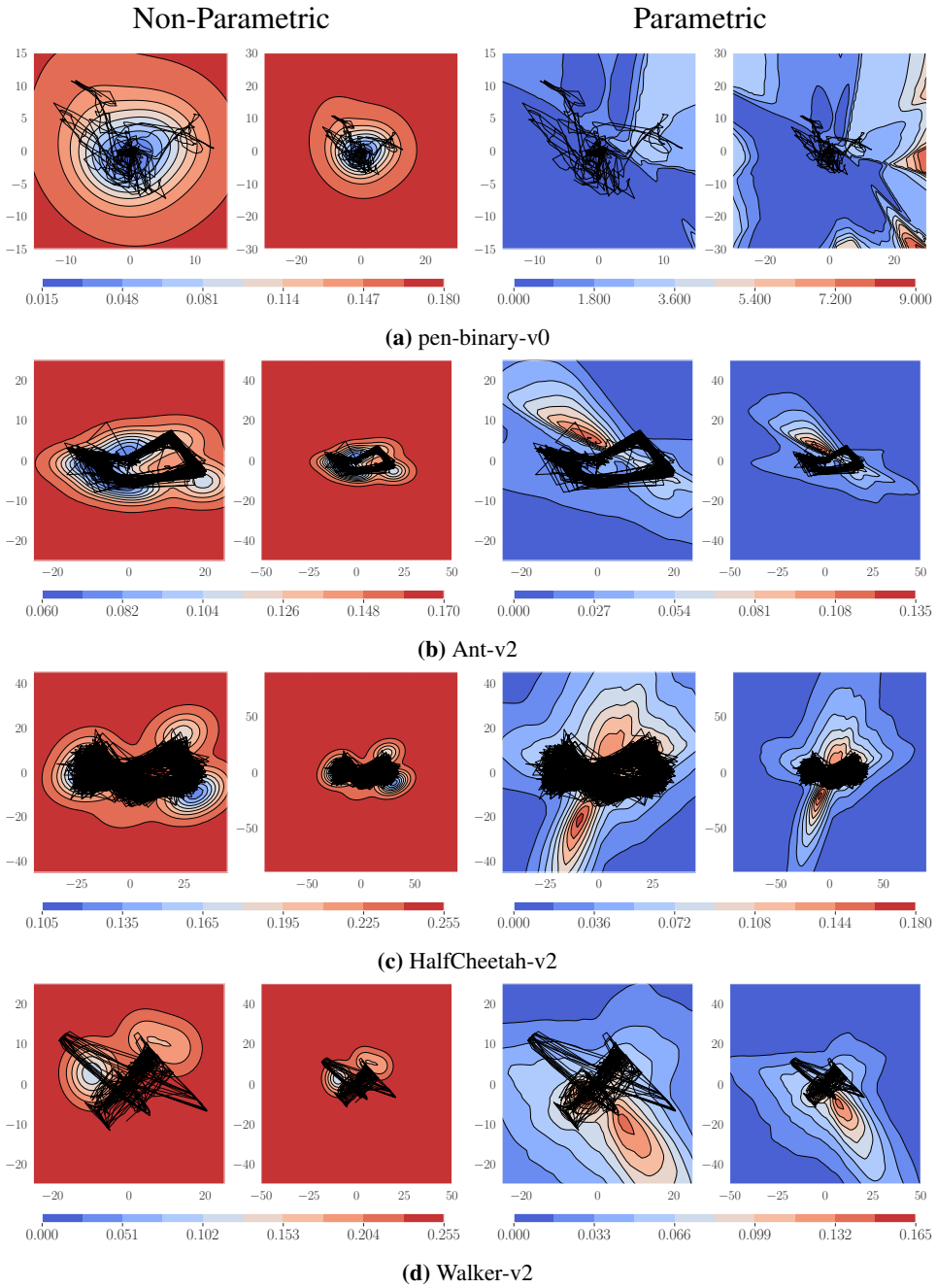
(a) Single Model **(b)** $K = 2$ **(c)** $K = 4$

(d) $K = 6$ **(e)** $K = 8$ **(f)** $K = 10$

Figure 7: Predictive variances of ensembles of parametric neural network Gaussian behavioral policies $\pi_{\psi^{1:K}}(\cdot \mid \mathbf{s})$ on a low-dimensional representation of the "door-binary-v0" environment, with each neural network in the ensemble trained via MLE. The ensemble policies are marginally better calibrated than parametric neural network policies in that their predictive variance only collapses in some but not all regions away from the expert trajectories.

E PREDICTIVE VARIANCE ESTIMATES ACROSS ENVIRONMENTS

Figure 8 shows the predictive variances of non-parametric and parametric behavioral policies on low dimensional representations of the environments considered in Figure 2 (excluding "door-binary-v0", which is shown in Figure 1).



(a) pen-binary-v0

(b) Ant-v2

(c) HalfCheetah-v2

(d) Walker-v2

Figure 8: Predictive variances of non-parametric and parametric behavioral policies on low dimensional representations of the environments considered in Figure 2 (excluding "door-binary-v0", which is shown in Figure 1). **Left Column**: 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 Column**: 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. As in Figure 1, the predictive variance of the GP is well-calibrated, whereas the predictive variance of the neural network is not.

F KL-REGULARIZED RL WITH NON-PARAMETRIC BEHAVIORAL POLICIES

Non-Degenerate Gaussian Process Behavioral Policies. Gaussian processes (GPs) (Rasmussen & Williams, 2005) 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 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, $\pi_0(\cdot \,|\, \mathbf{s})$ distributed according to a GP:

$$\mathbf{A} \,|\, \mathbf{s} \sim \pi_0(\cdot \,|\, \mathbf{s}) = \mathcal{GP}\big(m(\mathbf{s}), k(\mathbf{s}, \mathbf{s}')\big), \tag{F.21}$$

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 $\mathbf{A} \,|\, \mathbf{s}, \mathcal{D}_0 \sim \pi_0(\cdot \,|\, \mathbf{s}, \mathcal{D}_0) = \mathcal{GP}\big(\boldsymbol{\mu}_0(\mathbf{s}), \boldsymbol{\Sigma}_0(\mathbf{s}, \mathbf{s}')\big)$ and

$$\mu(\mathbf{s}) = m(\mathbf{s}) + k(\mathbf{s}, \bar{\mathbf{S}})(k(\bar{\mathbf{S}}, \bar{\mathbf{S}}))^{-1}(\bar{\mathbf{A}} - m(\bar{\mathbf{A}})) \tag{F.22}$$

$$\Sigma(\mathbf{s}, \mathbf{s}) = k(\mathbf{s}, \mathbf{s}) + k(\mathbf{s}, \bar{\mathbf{S}})(k(\bar{\mathbf{S}}, \bar{\mathbf{S}}))^{-1}k(\bar{\mathbf{S}}, \mathbf{s}). \tag{F.23}$$

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 can be scaled to $N > 1,000,000$ (Wang et al., 2019).

Figure 1 confirms that the non-parametric GP's predictive variance is well-calibrated when conditioned on expert trajectories: 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, Non-Parametric 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 (Hasselt, 2010). Before online training, the online policy is pre-trained to minimize the KL divergence to the behavioral policy on the offline dataset: $J_{\mathcal{GP}}(\phi) \stackrel{\text{def}}{=} \mathbb{E}_{\mathbf{s} \sim \mathcal{D}_0}\left[\mathbb{D}_{\text{KL}}(\pi_\phi(\cdot \,|\, \mathbf{s}) \,\|\, \pi_0(\cdot \,|\, \mathbf{s}))\right]$.

Algorithm 1 Non-Parametric Prior Actor–Critic

Input: offline dataset $\mathcal{D}_0$, initial parameters θ_1, θ_2, ϕ, GP $\pi_0(\cdot \,|\, \mathbf{s}) = \mathcal{GP}\big(m(\mathbf{s}), k(\mathbf{s}, \mathbf{s}')\big)$
Condition $\pi_0(\cdot \,|\, \mathbf{s})$ on $\mathcal{D}_0$ to obtain $\pi_0(\cdot \,|\, \mathbf{s}, \mathcal{D}_0)$
for each offline batch **do**
  $\phi \leftarrow \phi - \lambda_{\mathcal{GP}} \hat{\nabla}_\phi J_{\mathcal{GP}}(\phi)$ ▷ Minimize KL between online and behavioral policy.
end for

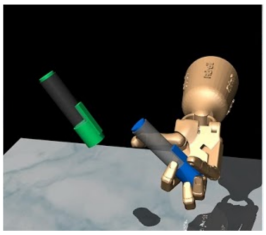
$\bar{\theta}_1 \leftarrow \theta_1, \bar{\theta}_2 \leftarrow \theta_2$ ▷ Initialize target network weights.
$\mathcal{D} \leftarrow \emptyset$ ▷ Initialize an empty replay pool.
for each iteration **do**
  for each environment step **do**
    $\mathbf{a}_t \sim \pi_\phi(\cdot \,|\, \mathbf{s}_t)$
    $\mathbf{s}_{t+1} \sim p(\cdot \,|\, \mathbf{s}_t, \mathbf{a}_t)$
    $\mathcal{D} \leftarrow \mathcal{D} \cup \{(\mathbf{s}_t, \mathbf{a}_t, r(\mathbf{s}_t, \mathbf{a}_t), \mathbf{s}_{t+1})\}$
  end for
  for each gradient step **do**
    $\theta_i \leftarrow \theta_i - \lambda_Q \hat{\nabla}_{\theta_i} J_Q(\theta_i)$ for $i \in \{1, 2\}$
    $\phi \leftarrow \phi - \lambda_\pi \hat{\nabla}_\phi J_\pi(\phi)$ ▷ Minimize $J_Q(\theta)$ and $J_\pi(\phi)$ using $\pi_0(\cdot \,|\, \mathbf{s}, \mathcal{D}_0)$.
    $\hat{\theta}_i \leftarrow \tau\theta_i + (1 - \tau)\hat{\theta}_i$ for $i \in \{1, 2\}$ ▷ Update target network weights.
  end for
end for
Output: Optimized parameters θ_1, θ_2, ϕ

G ENVIRONMENTS

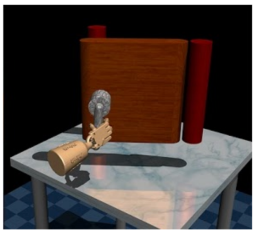
MuJoCo locomotion tasks. We evaluate our method on three representative tasks: Ant-v2, HalfCheetah-v2, and Walker2d-v2. For each task, we use 15 demonstration trajectories collected by a pre-trained expert, each containing 1000 steps.

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 (Rajeswaran et al., 2018b) 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. (2018b). 25 expert demonstrations were provided for each task Rajeswaran et al. (2018b), each consisting of 200 environment steps which are not fully optimal but do successfully solve the task.



(a) pen-binary-v0



(b) door-binary-v0

Figure 9: Visualizations of dexterous hand manipulation tasks.

H COMPARISON TO PRIOR WORK

To assess the usefulness of KL regularization for improving the performance and sample efficiency of online learning with expert demonstrations, we compare our approach to methods that incorporate expert demonstrations into online learning implicitly or explicitly via KL regularization as well as by means other than KL regularization.

ABM (Siegel et al., 2020). ABM explicitly KL-regularizes the online policy against a behavioral prior. This prior can be learned via MLE, like BRAC, or alternatively via an "advantage-weighted behavioral model" where the RL algorithm is biased to choose actions that are both supported by the offline data and that are good for the current task. This objective filters trajectory snippets by advantage-weighting, using an n-step advantage function. We show that no carefully chosen objective with additional hyperparameters is required.

AWAC (Nair et al., 2020). AWAC performs online fine-tuning of a policy pre-trained on offline. It achieves state-of-the-art results on the dexterous hand manipulation and MuJoCo continuous locomotion tasks. AWAC implicitly constrains the KL divergence of the online policy to be close to the behavioral policy by sampling from the replay buffer, which is initially filled with the offline data. The method requires additional off-policy data to be generated to saturate the replay buffer, thereby requiring a hidden number of environment interactions that do not involve learning. Our approach does not require the offline data to be added to the replay buffer before training.

AWR (Peng et al., 2019). AWR approximates constrained policy search by alternating between supervised value function and policy regression steps. The objective derived is similar to AWAC but instead estimates the value function of the behavioral policy which was demonstrated to be less efficient than Q-function estimation via bootstrapping (Nair et al., 2020). The method may be converted to use offline data by adding prior data to the replay buffer before training.

BEAR (Kumar et al., 2019). BEAR attempts to stabilize learning from off-policy data (such as offline data) by tackling bootstrapping error from actions far from the training data. This is achieved by searching for policies with the same support as the training distribution. This approach is too restrictive for the problem considered in this paper, since only a small number of expert demonstrations is available, which requires exploration. In contrast, our approach encourages exploration away from the data by wider prior predictive variances. BEAR uses an alternate divergence measure to the KL divergence, Maximum Mean Discrepancy (Smola et al., 2007). Other divergences such as Wasserstein Distances (Pacchiano et al., 2020) have also been proposed for regularization in RL.

BRAC (Wu et al., 2019). BRAC regularizes the online policy against an offline behavioral policy as our method does. However, BRAC exhibits the pathologies we have shown by learning a poor behavioral policy via MLE. To mitigate this, in practice, BRAC adds an entropy bonus to the supervised learning objective which stabilizes the variance around the training set but has no guarantees away from the data. We demonstrate that behavioral priors obtained via maximum likelihood estimation with entropy regularization exhibit a collapse in predictive uncertainty estimates way from the training data, resulting in the pathology described in Proposition 1.

DAPG (Rajeswaran et al., 2018a). DAPG incorporates offline data into policy gradients by initially pre-training with a behaviorally cloned policy and then augmenting the RL loss with a supervised-learning loss. We similarly pre-train the online policy at the start to avoid noisy KLs at the beginning of training. However, training a joint loss that combines two disparate and often divergent terms can be unstable.

SAC+BC (Nair et al., 2018) SAC+BC represents the approach of Nair et al. (2018) but uses SAC instead of DDPG (Lillicrap et al., 2016) as the underlying RL algorithm. The method maintains a secondary replay buffer filled with offline data that is sampled each update step, augmenting the policy loss with a supervised learning loss that is filtered by advantage and hindsight experience replay. Our method requires far fewer additional ad-hoc algorithmic design choices.

SACfD (Haarnoja et al., 2019). SACfD uses the popular Soft Actor–Critic (SAC) algorithm with offline data loaded into the replay buffer before online training. Our algorithm uses the same approximate policy iteration scheme as SAC with a modified objective. Nair et al. (2020) show that including the offline data into the replay buffer does not significantly improve the training performance over the unmodified SAC objective and that pre-training the online policy with offline data results in catastrophic forgetting. Thus, a different approach is needed to integrate offline data with SAC-style algorithms.

I HYPERPARAMETERS

Table 1 lists the hyperparameters used for N-PPAC. Multiple values refer to MuJoCo continuous control for the former and the dexterous hand manipulation tasks for the latter.

Table 1: N-PPAC Hyperparameters

Parameter	Value(s)
optimizer	Adam
learning rate	$3 \cdot 10^{-4}$
discount (γ)	0.99
reward scale	1
replay buffer size	10^6
number of hidden layers	$\{2, 4\}$
number of hidden units per layer	256
number of samples per minibatch	$\{256, 1024\}$
activation function	ReLU
target smoothing coefficient (τ)	0.005
target update interval	1
number of policy pretraining epochs	400
GP covariance function	{RBF, Matérn}

Table 2 lists the hyperparameters used to train the Gaussian process on the offline data. The hyperparameters are trained by maximizing the log-marginal likelihood.

Table 2: GP Optimization Hyperparameters

Parameter	Value
optimizer	Adam
learning rate	0.1
number of epochs	500

J ABLATION STUDY ON KL TEMPERATURE TUNING

Figure 10 shows that unlike in standard SAC (Haarnoja et al., 2019), tuning of the KL-temperature is not necessary to achieve a good online performance. Thus, we default to a fixed value for simplicity in our experiments.

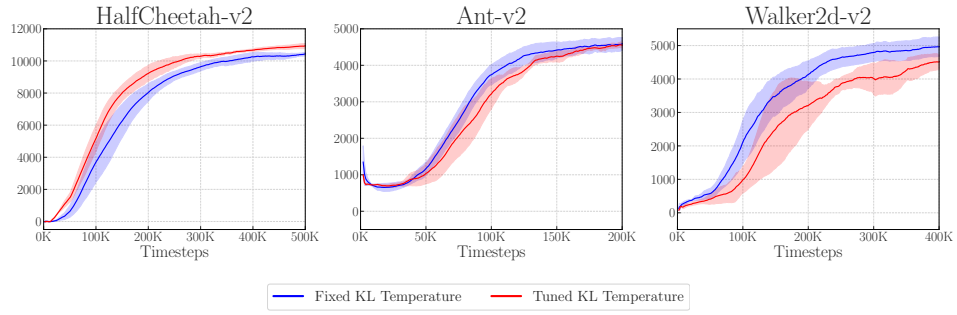


Figure 10: Ablation study on the effect of automatic KL temperature tuning on MuJoCo locomotion tasks.