LEARNING ROBUST CONTROLLERS VIA PROBABILISTIC MODEL-BASED POLICY SEARCH

Valentin Charvet, Bjørn Sand Jensen, and Roderick Murray-Smith
Department of Computing Science
University of Glasgow
v.charvet.1@research.gla.ac.uk

ABSTRACT

Model-based Reinforcement Learning estimates the true environment through a world model in order to approximate the optimal policy. This family of algorithms usually benefits from better sample efficiency than their model-free counterparts. We investigate whether controllers learned in such a way are robust and able to generalize under small perturbations of the environment. Our work is inspired by the PILCO algorithm, a method for probabilistic policy search. We show that enforcing a lower bound to the likelihood noise in the Gaussian Process dynamics model regularizes the policy updates and yields more robust controllers. We demonstrate the empirical benefits of our method in a simulation benchmark.

1 INTRODUCTION

Despite impressive ability at solving tasks such as video games (Mnih et al., 2013), *Reinforcement Learning (RL)* often fails to work well in real world situations. Dulac-Arnold et al. (2019) survey the causes for it and propose a set of nine challenges representing the difficulty to deploy agents outside of simulations. Among them is the problem of non-stationarity, which means the characteristics of the environment evolve during time. It is a similar issue to that of distribution shift, because the agent has not been trained in the new environment it interacts with. Interestingly, this problem can also be found in Batch RL where the policy is learned from a dataset of observations. We refer to the survey (Levine et al., 2020) for more details on that.

Learning robust policies is therefore a crucial step for deploying reinforcement learning agents. The reason is that over time, a robot's sensors can suffer from physical degradation which will impair their ability to relay the true state of the environment. Instead, they will only receive noisy estimations of it. Moreover, wear and tear are likely to modify the intrinsic physical nature of the agent or the environment itself. As a consequence, it is necessary to account for this variability during training.

The common ways to approach that problem include meta and adversarial learning. While the former learns a prior on predictive model from multiple environments the latter optimizes the policy for a pessimistic objective function.

The method we propose instead only requires one environment to be trained on and does not need to modify the objective function. We choose to build on PILCO by Deisenroth & Rasmussen (2011) because its use of probabilistic policy search shows natural robustness. We further regularize the Gaussian Process dynamics in order to increase its domain adaptation capabilities.A similar approach was applied by Igl et al. (2019) to deep neural networks. Finally we also integrate safety constraints to evaluate the robustness of our controllers more precisely.

2 RELATED WORK

The problem of domain adaptation is not exclusive to sequential decision-making problems and arises when training and evaluation datasets are not independents and identically distributed. One of the most classical approach to tackle this is by regularization (Hastie et al. (2009) chapters 3 to 6), a method that improves generalization by preventing overfitting. These tricks work well on both vision (Jeong & Shin, 2020; Balaji et al., 2018) and language (Balaji et al., 2018) tasks.

A promising direction of research is that of *meta-learning*. In (**?**Nagabandi et al., 2018), a prior on the system dynamics is learned during the initial training phase. When running in a new environment, this prior is updated so as to best reflect the new system's dynamics. This type of problem can also be solved by learning a hierarchical policy (Kupcsik et al., 2013), where the upper layer depends on the context and the lower one on the current state of the robot. Similarly (Yu et al., 2017) learns a unique policy in a several environment and use an identification method that feeds a context signal to the policy. The drawback of these methods is that they require access to a wide range of environment in order to learn a latent representation of the dynamics. In contrast, our method only needs access to one simulator.

Last, robustness can be achieved by optimizing a *pessimistic objective*. This is often referred to as the *Robust Markov Decision Process* (MDP) framework, which can be solved by approximate dynamics programming Mankowitz et al. (2018); Tamar et al. (2014) or within Maximum a Posteriori Policy Optimization Mankowitz et al. (2019). Such methods go as back as 2005 (Morimoto & Doya, 2005) where the authors apply an actor-critic where the controller attempts to correct for disturbances generated by an internal agent. More recently (Pinto et al., 2017) apply a similar method with neural networks. In essence, these methods solve a minimax optimization problem to account for worst-case scenarios.

3 METHODS

MODEL-BASED POLICY SEARCH

We consider the setting of an MDP (X, U, f, C, T) denoting respectively the state and action spaces, transition function, cost function and episode length T. The goal is, given a parametric *policy* (or *controller*) $\pi_{\boldsymbol{\theta}}$, to minimize the *return* or *cost-to-go* defined as the expected sum of future costs. In contrast, model-based RL optimizes the policy on the surrogate objective (1). In that equation, $\hat{f}$ is the estimate of f. It trained by minimizing the empirical risk between the samples $\hat{f}(\boldsymbol{x}_t^i, \boldsymbol{u}_t^i)$ and $\boldsymbol{x}_{t+1}^i$ encountered in the course of agent-environment interactions.

$$\boldsymbol{\theta}^* = \underset{\boldsymbol{\theta}}{\operatorname{argmin}}\, \mathbb{E}_{\pi_{\boldsymbol{\theta}}, \hat{f}} \left[\sum_{t=0}^{T-1} C(\boldsymbol{x}_t) \right]. \tag{1}$$

PILCO (Deisenroth & Rasmussen, 2011) is a model-based algorithm that updates its policy by predicting an approximate trajectory distribution over the episode length T. As such, it can be viewed as a model-based Monte-Carlo Policy Gradient method.

The core of the algorithm is the computation of the successor state distribution as $p(\boldsymbol{x}_{t+1}) = \int \hat{f}(\boldsymbol{x}_{t+1}|\boldsymbol{x}_t, \boldsymbol{u}_t) p(\boldsymbol{x}_t, \boldsymbol{u}_t) d\boldsymbol{x}_t d\boldsymbol{u}_t$ which is generally an intractable integral. It can be approximated via Moment Matching Girard et al. (2003), sampling (Gal et al., 2016; Parmas et al., 2018) or Numerical Quadrature (Vinogradska et al., 2020). The same method can then be used to approximate the local costs $c_t = \int C(\boldsymbol{x}_t) p(\boldsymbol{x}_t) d\boldsymbol{x}_t$, provided the cost function is in the polynomial or exponential family.

LEARNING WITH SAFETY CONSTRAINTS

In many situations, safety is of critical importance and needs to be incorporated in the policy search. We follow the same procedure as Polymenakos et al. (2019) but frame it as a *Primal-Dual* problem. This formulation has also been proposed in Cowen-Rivers et al. (2020) but its impact on robustness has not been evaluated. Let us consider a set of hazardous regions H that the agent must avoid. We consider the probability of being in that region a *chance constraint* written $G(\mathbf{z}) = \int_H p_{\boldsymbol{z}}(\boldsymbol{z}) d\boldsymbol{z}$.

The state-distribution approximation from the previous section (Moment Matching in our case) can be used to integrate the risk over the entire trajectory by computing $G(\boldsymbol{x}_t)$ for $t = 0 \dots T-1$. For a given threshold ξ we can then write the constrained objective as

$$\boldsymbol{\theta}^* = \underset{\boldsymbol{\theta}}{\arg\min} \quad \mathbb{E}_{\pi_{\boldsymbol{\theta}}, \hat{f}} \left[\sum_{t=0}^{T-1} C(\boldsymbol{x}_t) \right] \quad \text{subject to} \quad \mathbb{E}_{\pi_{\boldsymbol{\theta}}, \hat{f}} \left[\sum_{t=0}^{T-1} G(\boldsymbol{x}_t) \right] < \xi, \tag{2}$$

and the corresponding Lagrangian as

$$L(\boldsymbol{\theta}, \lambda) = V_{\boldsymbol{\theta}}(\boldsymbol{x}^0) + \lambda (Q_{\boldsymbol{\theta}}(\boldsymbol{x}^0) - \xi), \tag{3}$$

where $V_{\boldsymbol{\theta}}$ and $Q_{\boldsymbol{\theta}}$ are respectively the expected cost and risk-to-go from starting point $\boldsymbol{x}^0$. We can therefore find an optimal solution to the problem (2) by iteratively ascending the Lagrangian in λ and descending in $\boldsymbol{\theta}$.

ROBUSTNESS TO PERTURBATIONS

The strength of Gaussian Processes for planning is their ability to accurately model predictive uncertainty. However, GPs are not inherently robust to shift in the transition model thus predictions will not be valid when the evaluation environment differs from the training environment. We can mitigate this by forcing the model to be overly cautious by applying a lower bound on the inferred likelihood noise. Intuitively, if we force the learned noise to be greater than some value σ_{low} then the policy should be robust in an environment that has observational noise up to a level σ_{low}. The following simple analysis shows how it helps to generalize for small perturbations of latent parameters of the environment dynamics.

Assumption 1 (Smooth dynamics)**.** *We assume the true transition kernel of the system is a smooth function of its latent physical parameters $\boldsymbol{\phi}$ such that $\boldsymbol{x}' = f(\boldsymbol{x}, \boldsymbol{u}; \boldsymbol{\phi})$ and*

$$\|f(\boldsymbol{x}, \boldsymbol{u}; \boldsymbol{\phi}) - f(\boldsymbol{x}, \boldsymbol{u}; \boldsymbol{\phi} + \boldsymbol{\delta})\| \leq K\|\boldsymbol{\delta}\|, \qquad \forall \boldsymbol{x}, \boldsymbol{u}, \boldsymbol{\phi}, \boldsymbol{\delta} \tag{4}$$

where K is the Lipschitz constant and $\boldsymbol{\delta}$ is a perturbation.

Assumption 2. *The evaluation environment is a perturbed version of the trained one, it transitions according to $\boldsymbol{x}' = f(\boldsymbol{x}, \boldsymbol{u}; \boldsymbol{\phi}_0 + \boldsymbol{\delta})$, for a given $\boldsymbol{\delta}$ and where $\boldsymbol{\phi}_0$ are the parameters of the training environment.*

Lemma 3.1 (Bound on physical perturbations)**.** *Under assumptions 1 and 2, enforcing a lower bound on the noise of the GP model enables optimizing the controller for all the environments in a ball centered at $\boldsymbol{\phi}_0$ of radius $\boldsymbol{\delta}_{max}$, with $\|\boldsymbol{\delta}_{max}\| \leq \sigma_{low}/K$.*

Proof. The detail of the derivation is available in appendix A. □

The first conclusion we draw from lemma (3.1) is that the algorithm should be able to generalize in environments similar enough to the training one. The lower K is, the easier it is to generalize. This behavior is what we would expect intuitively, as a low value of K means the environment dynamics vary little with its latent parameters. In theory, Lemma (3.1) means we can generalize in a region with any radius we want by choosing a high value for σ_{lbound}. However, because PILCO predictions are autoregressive, a high value of the noise bound will lead to high variance in the predicted trajectory.

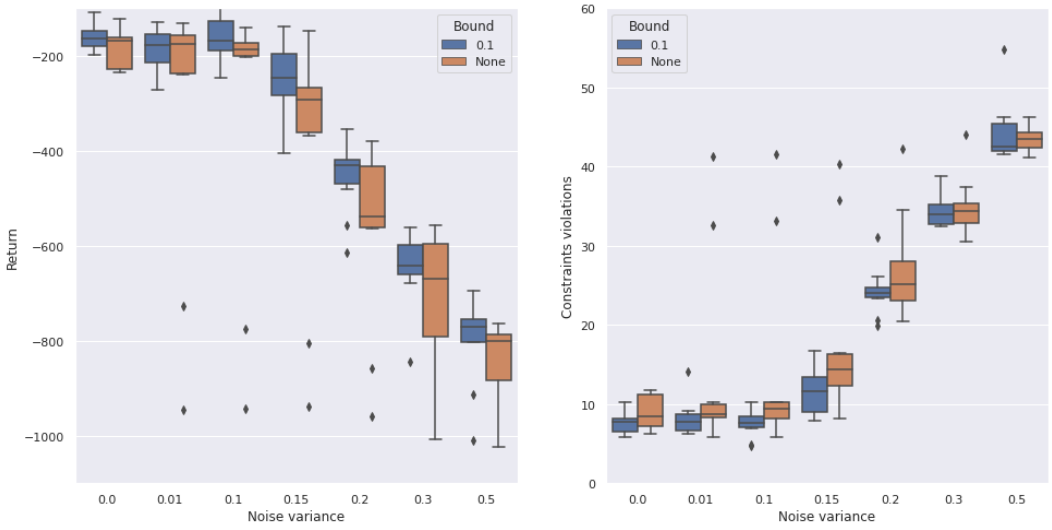

Figure 1: Return (left) and risk (right) for $\sigma_{low} = 0.1$ and with no bound. The dots represent values outside the inter-quartiles range (see figure in appendix B for the full graph).

σ_{low}	$\sigma_{perturb}$ 0	10^{-2}	10^{-1}	0.15	0.2
None (no safety)	-302 ± 1506 14 ± 85	-346 ± 192 14 ± 15	-341 ± 185 17 ± 12	-501 ± 263 23 ± 18	-630 ± 157 31 ± 14
None	-166 ± 1736 8 ± 78	-174 ± 285 8 ± 12	-186 ± 290 9 ± 12	-291 ± 259 14 ± 10	-538 ± 191 25 ± 10
10^{-3}	-197 ± 1208 8 ± 64	-212 ± 206 8 ± 11	-209 ± 222 9 ± 11	-379 ± 202 11 ± 9	-572 ± 146 26 ± 7
10^{-2}	-213 ± 1060 $\mathbf{7 \pm 5}$	-185 ± 190 8 ± 9	-198 ± 202 8 ± 8	-272 ± 180 12 ± 8	-518 ± 112 23 ± 4
0.1	$\mathbf{-163 \pm 26}$ $\mathbf{7 \pm 1}$	$\mathbf{-177 \pm 45}$ $\mathbf{7 \pm 2}$	$\mathbf{-167 \pm 46}$ $\mathbf{7 \pm 1}$	-247 ± 75 11 ± 3	$\mathbf{-420 \pm 76}$ 24 ± 3
0.2	-171 ± 50 $\mathbf{7 \pm 1}$	-184 ± 34 $\mathbf{7 \pm 1}$	-176 ± 144 8 ± 6	$\mathbf{-235 \pm 207}$ $\mathbf{10 \pm 9}$	-430 ± 169 $\mathbf{22 \pm 5}$

Table 1: Median and standard deviation values for performance on the pendulum, top row of each cell is the return (higher is better) and bottom row is number of constraints violations (lower is better). We highlight values with good median performance *and* low variance

This can be alleviated in practice by rolling out the state predictions on short horizons, similarly as in Janner et al. (2019). In practice, the constant K can be estimated with classical tools (Wood & Zhang, 1996) but it requires access to versions of the system with different latent parameters. Alternatively to the soft constraint we propose, a prior on the hyperparameter as in Burnaev et al. (2016) acting as a regularization of the evidence maximisation procedure could improve robustness of the policy search. We leave investigation of such priors to future work.

4 Experiments

The question the experiments aim to answer is whether a lower bound of the noise is a good enough regularization to learn robust policies. The experimental setup and hyperparameters are detailed in appendix B. To evaluate the robustness of our controller to environmental perturbations, we first train the controller on the noise-free environment. We then roll out the controller in the perturbed evaluation environment and record the average returns and risks in 10 runs. We perturb the environment with observation noise as well as variation of the system's parameters, with global variance of perturbation $\sigma_{perturb}$. We show the evaluation performance for a given bound in Figure 1. All the results are obtained with models trained from 10 different random seeds.

Table 1 summarizes the experiment results, with PILCO without safety constraints as baseline. The first row is obtained with basic PILCO algorithm without safety constraint. We achieve more consistent results with higher values of the bound 0.1 and 0.2 hence showing the controllers are more robust. Moreover the values with a bound of 0.1 and 0.2 have much less variance than the rest. This means the said models have converged for all 10 random initializations. This is confirmed by figure 1, where the distributions of the performance with no bound have many more outliers, that is runs that have not successfully converged. We can see on table 1 the trade-off between a high bound that improves generalization and a low one that prevent high variance in the predicted trajectories. Therefore, the most significant improvement of our method is variance reduction: a real benefit as it indicates a more stable training.

5 Conclusion

In this paper, we proposed a method to increase the robustness of controllers trained with Model-Based Reinforcement Learning by using a regularization on the Gaussian Process noise hyperparameter. We empirically demonstrated a significant decrease in performance fluctuations for higher noise bounds, hence improving domain adaptation. We also believe these ideas can be applied to the offline RL case and plan to investigate this direction in future work.

ACKNOWLEDGEMENTS

This work is supported by EPSRC Grant EP/R018634/1 (Closed-Loop Data Science for Complex, Computationally- and Data-Intensive Analytics) and the University of Glasgow. We thank the contributors open-source software libraries we used for this work, specifically GPyTorch (Gardner et al., 2018), PyBullet Gym (Ellenberger, 2018–2019) and Seaborn (Waskom & the seaborn development team, 2020).

REFERENCES

Yogesh Balaji, Swami Sankaranarayanan, and Rama Chellappa. Metareg: Towards domain generalization using meta-regularization. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 31, 2018.

E. V. Burnaev, M. E. Panov, and A. A. Zaytsev. Regression on the basis of nonstationary Gaussian processes with Bayesian regularization. *Journal of Communications Technology and Electronics*, 61(6):661–671, jun 2016. ISSN 10642269. doi: 10.1134/S1064226916060061.

Alexander I. Cowen-Rivers, Daniel Palenicek, Vincent Moens, Mohammed Abdullah, Aivar Sootla, Jun Wang, and Haitham Ammar. Samba: Safe model-based active reinforcement learning. 2020.

Marc Peter Deisenroth and Carl Edward Rasmussen. PILCO: A Model-Based and Data-Efficient Approach to Policy Search. In *International Conference on Machine Learning*. 467-472, 2011. doi: 10.1080/0034408960910404.

Gabriel Dulac-Arnold, Daniel Mankowitz, and Todd Hester. Challenges of real-world reinforcement learning. 2019.

Benjamin Ellenberger. Pybullet gymperium. `https://github.com/benelot/pybullet-gym`, 2018–2019.

Yarin Gal, Rowan Thomas Mcallister, and Carl Edward Rasmussen. Improving PILCO with Bayesian Neural Network Dynamics Models. *Data-Efficient Machine Learning Workshop, ICML*, pp. 1–7, 2016.

Jacob Gardner, Geoff Pleiss, Kilian Q Weinberger, David Bindel, and Andrew G Wilson. Gpytorch: Blackbox matrix-matrix gaussian process inference with gpu acceleration. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 31, 2018.

Agathe Girard, Carl Edward Rasmussen, Joaquin Quinonero-Candela, and Roderick Murray-Smith. Gaussian process priors with uncertain inputs - application to multiple-step ahead time series forecasting. 2003.

Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Second Edition*, volume 77. 2009. doi: 10.1111/j.1751-5823.2009.00095_18.x.

Maximilian Igl, Kamil Ciosek, Yingzhen Li, Sebastian Tschiatschek, Cheng Zhang, Sam Devlin, and Katja Hofmann. Generalization in reinforcement learning with selective noise injection and information bottleneck. 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.

Michael Janner, Justin Fu, Marvin Zhang, and Sergey Levine. When to trust your model: Model-based policy optimization. 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.

Jongheon Jeong and Jinwoo Shin. Consistency regularization for certified robustness of smoothed classifiers. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 10558–10570, 2020.

Andras Kupcsik, Marc Deisenroth, Jan Peters, and Gerhard Neumann. Data-efficient generalization of robot skills with contextual policy search. In *Proceedings of the AAAI conference on artificial intelligence*, volume 27, 2013.

Sergey Levine, Aviral Kumar, George Tucker, and Justin Fu. Offline Reinforcement Learning: Tutorial, Review, and Perspectives on Open Problems. May 2020.

Daniel Mankowitz, Timothy Mann, Pierre-Luc Bacon, Doina Precup, and Shie Mannor. Learning robust options. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

Daniel J Mankowitz, Nir Levine, Rae Jeong, Abbas Abdolmaleki, Jost Tobias Springenberg, Yuanyuan Shi, Jackie Kay, Todd Hester, Timothy Mann, and Martin Riedmiller. Robust reinforcement learning for continuous control with model misspecification. In *International Conference on Learning Representations*, 2019.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. 2013.

Jun Morimoto and Kenji Doya. Robust reinforcement learning. *Neural computation*, 17(2):335–359, 2005.

Anusha Nagabandi, Ignasi Clavera, Simin Liu, Ronald S Fearing, Pieter Abbeel, Sergey Levine, and Chelsea Finn. Learning to adapt in dynamic, real-world environments through meta-reinforcement learning. *arXiv preprint arXiv:1803.11347*, 2018.

Paavo Parmas, Carl Edward Rasmussen, Jan Peters, and Kenji Doya. PIPPS: Flexible model-based policy search robust to the curse of chaos. In *35th International Conference on Machine Learning, ICML 2018*, volume 9, pp. 6463–6472, jul 2018. ISBN 9781510867963.

Lerrel Pinto, James Davidson, Rahul Sukthankar, and Abhinav Gupta. Robust adversarial reinforcement learning. In *International Conference on Machine Learning*, pp. 2817–2826, 2017.

Kyriakos Polymenakos, Alessandro Abate, and Stephen Roberts. Safe policy search using gaussian process models. In *Proceedings of the 18th International Conference on Autonomous Agents and Multi Agent Systems*, pp. 1565–1573, 2019.

Aviv Tamar, Shie Mannor, and Huan Xu. Scaling up robust mdps using function approximation. In Eric P. Xing and Tony Jebara (eds.), *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pp. 181–189, Bejing, China, 22–24 Jun 2014.

Julia Vinogradska, Bastian Bischoff, Jan Achterhold, Torsten Koller, and Jan Peters. Numerical Quadrature for Probabilistic Policy Search. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(1):164–175, 2020. ISSN 19393539. doi: 10.1109/TPAMI.2018.2879335.

Michael Waskom and the seaborn development team. mwaskom/seaborn, September 2020. URL `https://doi.org/10.5281/zenodo.592845`.

G.R. Wood and B.P. Zhang. Estimation of the Lipschitz constant of a function. *Journal of Global Optimization*, 8(1):91–103, 1996.

Wenhao Yu, Jie Tan, C Karen Liu, and Greg Turk. Preparing for the unknown: Learning a universal policy with online system identification. *arXiv preprint arXiv:1702.02453*, 2017.

A PROOF: BOUND ON PHYSICAL PARAMETERS

Proof. We show the bound in the case where the system is one dimensonal. In the following we omit the dependency of the dynamics on the control signal for clarity.

Suppose we have trained the estimate of the dynamics on the noise-free system $\boldsymbol{\phi}_0$, and enforced a bound on the learned likelihood noise σ_{low}. We write the transitions of the true dynamics $f(\boldsymbol{x}, \boldsymbol{\phi}_0)$ and the estimate learned from it as $\hat{f}(\boldsymbol{x}, \boldsymbol{\phi}_0)$.

Therefore we have

$$\hat{f}(\boldsymbol{x}, \boldsymbol{\phi}_0) \leq f(\boldsymbol{x}, \boldsymbol{\phi}_0) + \sigma_{low} \ , \qquad \text{with high probability.} \tag{5}$$

Applying the Lipschitz continuity assumption stated in Eq. (4)

$$\begin{aligned} f(\boldsymbol{x}, \boldsymbol{\phi}_0 + \boldsymbol{\delta}) &\leq f(\boldsymbol{x}, \boldsymbol{\phi}_0) + K\|\boldsymbol{\delta}\| \\ &\leq \hat{f}(\boldsymbol{x}, \boldsymbol{\phi}_0) + K\|\boldsymbol{\delta}\| - \sigma_{low} \qquad \text{using (5)} \end{aligned} \tag{6}$$

We can write a lower bound for $f(\boldsymbol{x}, \boldsymbol{\phi}_0 + \boldsymbol{\delta})$ similarly as Eq. (6) and therefore

$$\left| f(\boldsymbol{x}, \boldsymbol{\phi}_0 + \boldsymbol{\delta}) - \hat{f}(\boldsymbol{x}, \boldsymbol{\phi}_0) \right| \leq |K\|\boldsymbol{\delta}\| - \sigma_{low}| \tag{7}$$

The right-hand-side of equation (7) is a convex function of $\boldsymbol{\delta}$ and is minimized and equal to 0 for $\boldsymbol{\delta} \in \{\boldsymbol{\delta}', \ K\|\boldsymbol{\delta}'\| < \sigma_{low}\}$, which concludes the proof. □

B DETAILS OF THE EXPERIMENTAL SETUP

HYPERPARAMETERS AND SETUP

We use the pendulum benchmark environment for our experiments. We learn the dynamics with a Sparse Gaussian Process model and the policy is parameterized by a nonlinear Radial Basis Function network. We plan long term trajectories with Moment Matching approximations and both policy parameters and dual variable are optimized via Gradient Descent. We chose to extend the classic PILCO algorithm with safety constraints as they are one of the challenges defined in (Dulac-Arnold et al., 2019).

The hyperparameters of the algorithm are shown in table 2. We use the same learning rates, optimisation algorithm and epochs for updating the controller and model at each episode. The value λ_0 is the initial value for the dual variable. The constraint we choose essentially prevents the pendulum from swinging-up the mass from its left side. We use a saturating cost function, centered around the target (pendulum at upright position). The parameters of the pendulum we are perturbing are: pole length, mass and gravity vector.

Table 2: Hyperparameters Summary

Parameter	**Value**
Inducing Points	100
Number of basis functions	50
Prediction Horizon	40
Number of episodes	20
Optimiser	Adam
Learning Rate	0.01
Epochs per episode	100
Constraint	$\theta \in [45, 135]$
ξ	1
λ_0	20

NOTE ON LOWER BOUND CONSTRAINTS

In order to impose a lower bound on the inferred noise, we optimize that hyperparameter in a transformed space. Therefore we have $\sigma_{raw} \in \mathbb{R}$, optimized so as to minimize the negative log likelihood. For predictions, this value is transformed so as to be greater than σ_{low} with $\sigma_{noise} = SoftPlus(\sigma_{raw}) + \sigma_{low}$

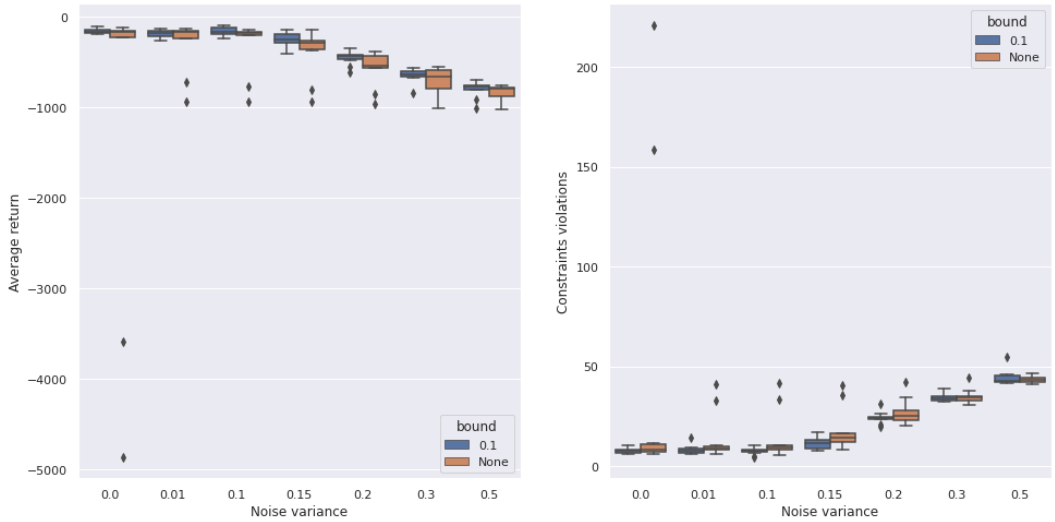

Figure 2: Return (left) and risk (right) with $\sigma_{\text{lbound}} = 0.1$ and no bound. The dots represent values outside the inter-quartiles range