Learning a Risk-Averse Locomotion Policy from Uncertainty Estimates

Thomas B. Brunner*, Takahiro Miki*, Josip Josifovski, Joonho Lee, Alois Knoll, Marco Hutter *†‡

Abstract: Recent advances in deep reinforcement learning have led to impressive progress in the field of autonomous robotic systems. These systems, however, often display overconfident behavior when interacting with the environment, as they are not aware of the uncertainty in their predictions. In this work, we present an approach of incorporating uncertainty into a learning-based locomotion controller for a quadrupedal robot. For this, we estimate the uncertainty in the internal latent state of the controller, which is then used by the controller to generate risk-averse behavior. Our approach consists of training an initial policy using a supervised loss based on demonstrations from an expert. During this stage, we also learn estimates of the uncertainty in the internal belief of the controller using ground truth data from a simulated training environment. The policy is then fine-tuned using reinforcement learning to learn risk-averse behavior given the estimate of its own uncertainty. Our results indicate that this approach leads to more careful behavior under high-uncertainty both in simulation and on a real robot. Link to videos: thomasbbrunner.github.io/learning-risk-averse-locomotion

1 Introduction

Deep reinforcement learning has brought significant advances to the field of robotics, where it has been applied for legged locomotion [1, 2], dexterous manipulation [3] and navigation [4]. In spite of these advances, acting safely and robustly in complex real-world environments is still an open challenge. Several works have presented methods for estimating uncertainties in the predictions from neural networks [5, 6, 7, 8], especially in the context of computer vision. Recent publications [9, 10, 11] have attempted to improve robustness of robotic systems by using estimates of the uncertainty in the prediction of a neural network. However, these approaches were often limited to simple synthetic tasks and were not shown in real-world applications. In this work, we present a method of incorporating uncertainty into the task of robotic locomotion with the goal of improving the safety and robustness of mobile robots.

This work is based on previous research done with the ANYmal robot [12], which demonstrated the robot's capabilities in navigating over rough terrain, operating under challenging conditions, and autonomously dealing with a variety of disturbances [13, 12, 2] by using privileged learning [14]. In this approach, a teacher policy is trained using ground truth data from the simulation and then the teacher's behavior is distilled into a student policy, which has access only to the observations available to the real robot. This enables a robust policy which can work in the real-world environment.

However, in this approach, the student policy is limited to imitating the behavior of the teacher policy which behaves confidently as it has ground truth observations. For instance, when faced with terrain observation heavily corrupted by noise, the student does not adjust its behavior, even though it cannot foresee the terrain in front. Ideally, the policy should behave more carefully in a highly uncertain situation.

*: equal contribution

† T. B. Brunner, T. Miki, J. Lee and M. Hutter are with the Robotic Systems Lab, ETH Zurich, Switzerland.

‡ T. B. Brunner, J. Josifovski and A. Knoll are with the Department of Computer Engineering, Technical University of Munich, Germany.

8th Conference on Robot Learning (CoRL 2024), Munich, Germany.

This work tackles these deficiencies by explicitly estimating both data and model uncertainties [5] in the locomotion controller, and by encouraging it to use these estimates to learn risk-averse behavior. Our method is evaluated both in simulation and in a real robot system. The results show that the controller is able to identify uncertainty and act differently under these situations.

In summary, the main contributions of our work are:

1. proposing an approach to estimate both model and data uncertainties by leveraging the ground truth data from simulated training environments, and
2. showing that the uncertainty estimates can be leveraged by RL agents to generate risk-averse behavior.

2 Methods

In this work, we extend the methodology proposed in [12] with the aim of addressing limitations both in current learning-based locomotion approaches and current methods of using uncertainty estimates in RL applications. First, we make the simulation environment more challenging by adding non-traversable terrain and a more expressive modeling of observation noise. In a second step, we implement methods to explicitly estimate the uncertainty in the internal belief of the controller. Lastly, we propose a training approach which leverages the modeled uncertainty to learn risk-averse behavior. Similarly to [12], the proprioceptive observations of the controller are composed of joint states, joint velocity histories, action history, leg phases, and the desired locomotion command. The exteroceptive inputs are comprised of samples from a robot-centered elevation map. During training, the controller also has access to privileged observations from the simulation environment, such as contact states, friction coefficients, and external disturbances. These observations cannot be measured in the real-world, but are used to facilitate training of the policy in the simulation.

2.1 Training Environment

Locomotion controllers face a variety of challenges in the real-world that lead to noisy sensor readings. To correctly predict the data uncertainty in its observations, the controller has to experience a diverse set of noise patterns in the simulation. To ensure this, we perturb the exteroceptive observations using a randomly generated noise configuration. The simulated environment is randomly divided into subregions and for each region a different noise configuration is sampled.

2.2 Uncertainty Estimation

Our controller architecture can be divided into three subnetworks: the belief encoder, the belief decoder and the action network, as seen in Figure 1.

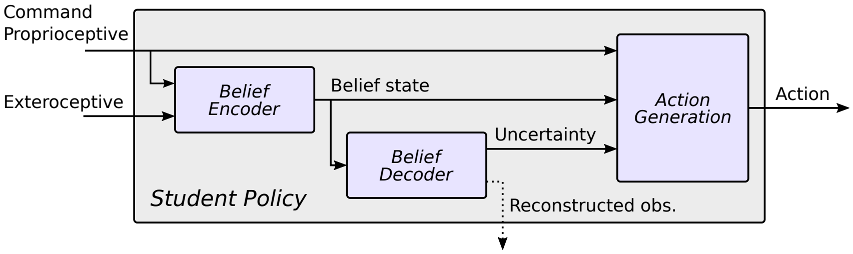

Figure 1: Overview of the architecture of the student policy network. The action generation module takes the raw proprioceptive inputs, the predicted belief state and the estimated uncertainty as inputs to predict actions. The reconstructed observations of the belief decoder are not directly used by the action module.

We model the data uncertainty by predicting an input-dependent variance estimate [5, 8]. The decoder predicts reconstructed observations, which can be compared with the ground truth observations to implicitly train the data uncertainty, as described in Equation 1. We follow the approach of training a neural network $\boldsymbol{f}$ parametrized by $\boldsymbol{\theta}$ to predict both a mean $\hat{\boldsymbol{y}}$ and also the corresponding variance[4] of the output $\boldsymbol{\sigma}^2$, which is a measure of the data uncertainty [5, 8] . The corresponding loss function [5, 8] for training the network given N training samples x_i, y_i is:

$$L_{\text{data}}(\boldsymbol{\theta}) = \frac{1}{N}\sum_{i}^{N} \frac{1}{2}\sigma_i^{-2}\left(\hat{y}_i - y_i\right)^2 + \frac{1}{2}\log\sigma_i^2 \tag{1}$$

To estimate the model uncertainty, we combine two approaches: Monte Carlo dropout [7] and deep ensembles [8]. Given the estimation of data and model uncertainties described above, we can combine them into a single approach. The inference procedure for each forward pass $m \in [1, ..., M]$ then becomes:

$$\hat{y}_m, s_m = f_{\boldsymbol{\theta}_m}(x) \quad \text{with} \quad s := \log\sigma^2 \quad \text{and} \quad \boldsymbol{\theta}_m \sim q(\boldsymbol{\theta}) \tag{2}$$

The final prediction can be computed by taking the mean of the predictions of all M forward passes. To compute the total uncertainty in the prediction, we sum the model uncertainty with the average data uncertainty of the forward passes:

$$\text{Var}(\hat{y}) \approx \underbrace{\frac{1}{M}\sum_{m}^{M}\hat{y}_m^2 - \left(\frac{1}{M}\sum_{m}^{M}\hat{y}_m\right)^2}_{\text{model}} + \underbrace{\frac{1}{M}\sum_{m}^{M}\sigma_m^2}_{\text{data}} \tag{3}$$

2.3 Learning Risk-Averse Behavior

In the next step, we leverage the uncertainty estimates to learn risk averse locomotion behavior. To encourage the student policy to develop its own behavior, we propose to add another step to the training procedure, which consists of fine-tuning the student policy using a RL setup similar to that of the teacher policy. Therefore, our objective during this step is to find a policy that maximizes the cumulative reward of the task R given noisy observations $\tilde{o}_t$, the belief state b_t and the uncertainty in the belief Var(b_t):

$$\max_{\pi} \mathbb{E}[R \mid \pi(\tilde{o}_t, b_t, \text{Var}(b_t))] \tag{4}$$

The complete approach is composed, thus, of three steps. First, a teacher policy is trained using RL and privileged learning. The near-optimal behavior learned by the teacher is then transferred to the student policy using supervised behavior cloning [14]. Finally, the student policy is fine-tuned using RL to encourage the learning of robust behavior under high uncertainty. An overview of the training steps can be seen in Figure 7.

3 Result

To validate the proposed approach, we first investigated the performance of the controller in synthetic experiments in the simulation. Our evaluation is focused on two aspects: (i) determining if the controller is able to learn risk-averse behavior, and (ii) determining if an explicit modeling of the uncertainty is beneficial for the aforementioned goal. For this, two policies are trained, one that uses explicit estimation of the uncertainty in the exteroceptive observations, and another policy that uses the same architecture, but without any notion of uncertainty (baseline). Both policies are pre-trained using imitation learning from a teacher policy and fine-tuned using RL.

We evaluated our method in simulated experiments where the robot navigates over flat terrain towards a goal position with a region of uncertainty in the middle. Each policy is evaluated a total of $N_T = 20$ times for different noise intensities. We count the number of times N_R that the robot refuses to advance into the region of uncertainty and compute a measure of relative risk-aversity

[4] To improve training stability, a common approach is to predict the log variance $\boldsymbol{s} := \log\boldsymbol{\sigma}^2$ instead [5].

$R = N_R/N_T$. A high value of R represents risk-averse behavior characterized by identifying and avoiding the region of uncertainty.

The result indicates that fine-tuning with RL was necessary to generate risk-averse behavior (Figure 2(a)). Moreover, using explicit uncertainty representation improved the risk-sensitivity as shown in Figure—2(b) where the policy could avoid the uncertain region with lower noise intensity. Examples of the behaviors are: reducing the walking velocity, taking smaller steps, avoiding the region of uncertainty, and even stepping backwards to move away from the uncertain region.

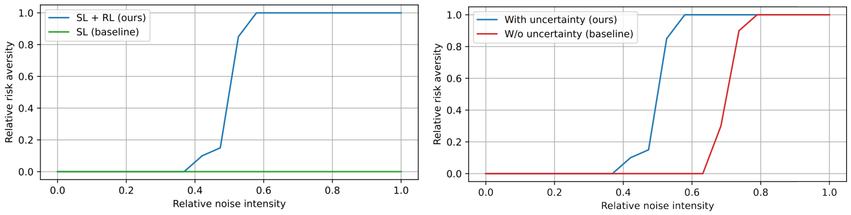

Figure 2: Risk-averse evaluation in simulation. Plots show how risk-averse a policy is for different noise intensities. The lowest noise intensity value represents no noise, while the highest stands for the highest noise intensity used during training. Left shows that fine-tuning with RL was necessary for showing the risk-averse behavior. Right shows the effectiveness of using explicit uncertainty representation.

3.1 Experiments on Real Robot

We evaluated our policy's effectiveness on real hardware through tests involving uncertain areas. First, we placed the robot on a box, where the edges became unobserved due to occlusion, creating gaps in the elevation map. In this scenario, our policy successfully prevented the robot from stepping off the box, as it recognized the surrounding region as uncertain. Next, we covered the front depth camera with tape, making the area in front of the robot unknown. Consequently, our policy refused to walk forward until the area became visible via the side cameras. Once observed, the robot could proceed safely.

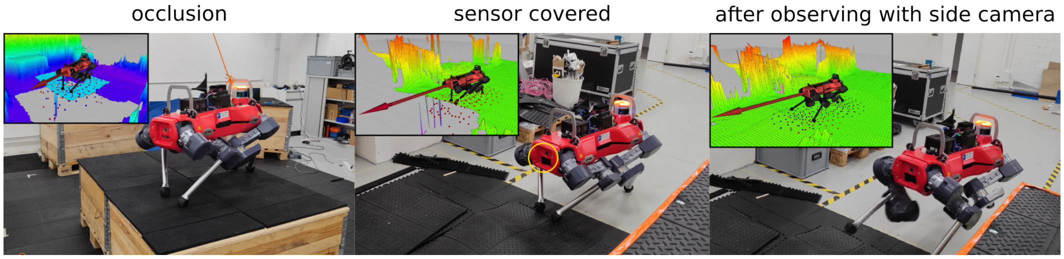

Figure 3: Evaluation of the proposed controller on the ANYmal-D robot. Subfigure shows the robot's heightmap, the sampled height scans (red and blue dots) and the input command (red arrow). Due to sensor occlusion, the controller has no information beyond the edge of the box, which leads to high uncertainty in the controller's internal belief state. To avoid the uncertain region, the controller refuses to follow the operator's command.

4 Conclusion

In this work, we developed a method to train a robot locomotion controller that generates risk-averse behavior using explicit uncertainty estimates and fine-tuning with reinforcement learning. We demonstrated the effectiveness of this approach in both simulation and real-world tests on the ANYmal-D robot, which successfully avoided uncertain areas. Future work will be enabling active perception, allowing the robot not only to act cautiously but also to turn and observe unknown areas before proceeding, rather than simply stopping.

References

[1] J. Hwangbo, J. Lee, A. Dosovitskiy, D. Bellicoso, V. Tsounis, V. Koltun, and M. Hutter. Learning agile and dynamic motor skills for legged robots. *Science Robotics*, 4(26):eaau5872, 2019.

[2] M. Tranzatto, F. Mascarich, L. Bernreiter, C. Godinho, M. Camurri, S. Khattak, T. Dang, V. Reijgwart, J. Loeje, D. Wisth, et al. Cerberus: Autonomous legged and aerial robotic exploration in the tunnel and urban circuits of the darpa subterranean challenge. *arXiv preprint arXiv:2201.07067*, 2022.

[3] OpenAI, I. Akkaya, M. Andrychowicz, M. Chociej, M. Litwin, B. McGrew, A. Petron, A. Paino, M. Plappert, G. Powell, R. Ribas, J. Schneider, N. Tezak, J. Tworek, P. Welinder, L. Weng, Q. Yuan, W. Zaremba, and L. Zhang. Solving rubik's cube with a robot hand, 2019. URL `https://arxiv.org/abs/1910.07113`.

[4] G. Kahn, A. Villaflor, B. Ding, P. Abbeel, and S. Levine. Self-supervised deep reinforcement learning with generalized computation graphs for robot navigation, 2017. URL `https://arxiv.org/abs/1709.10489`.

[5] A. Kendall and Y. Gal. What uncertainties do we need in bayesian deep learning for computer vision? *Advances in neural information processing systems*, 30, 2017.

[6] Y. Gal and Z. Ghahramani. A theoretically grounded application of dropout in recurrent neural networks, 2015. URL `https://arxiv.org/abs/1512.05287`.

[7] Y. Gal and Z. Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pages 1050–1059. PMLR, 2016.

[8] B. Lakshminarayanan, A. Pritzel, and C. Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles, 2016. URL `https://arxiv.org/abs/1612.01474`.

[9] B. Lütjens, M. Everett, and J. P. How. Safe reinforcement learning with model uncertainty estimates, 2018. URL `https://arxiv.org/abs/1810.08700`.

[10] G. Kahn, A. Villaflor, V. Pong, P. Abbeel, and S. Levine. Uncertainty-aware reinforcement learning for collision avoidance, 2017. URL `https://arxiv.org/abs/1702.01182`.

[11] A. Loquercio, M. Segu, and D. Scaramuzza. A general framework for uncertainty estimation in deep learning. *IEEE Robotics and Automation Letters*, 5(2):3153–3160, apr 2020. doi: 10.1109/lra.2020.2974682. URL `https://doi.org/10.1109%2Flra.2020.2974682`.

[12] T. Miki, J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, and M. Hutter. Learning robust perceptive locomotion for quadrupedal robots in the wild. *Science Robotics*, 7(62):eabk2822, 2022.

[13] J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, and M. Hutter. Learning quadrupedal locomotion over challenging terrain. *Science robotics*, 5(47):eabc5986, 2020.

[14] D. Chen, B. Zhou, V. Koltun, and P. Krähenbühl. Learning by cheating. In *Conference on Robot Learning*, pages 66–75. PMLR, 2020.

5 Appendix

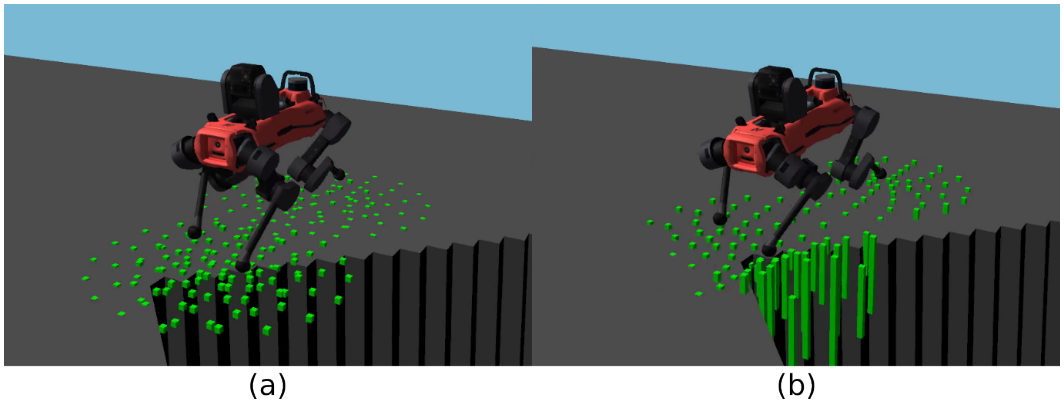

Figure 4: Robot approaching a pit, which cannot be observed due to sensor occlusion. Left image illustrates the internal belief of the controller without any uncertainty estimates. The controller assumes flat terrain and has no means of quantifying its confidence in that prediction. Right image shows an uncertainty-aware policy. The predicted belief is still flat, however, the policy is able to correctly estimate the uncertainty in the belief states (represented by the elongated height scans for the forward left leg and, to a lesser degree, for the forward right leg).

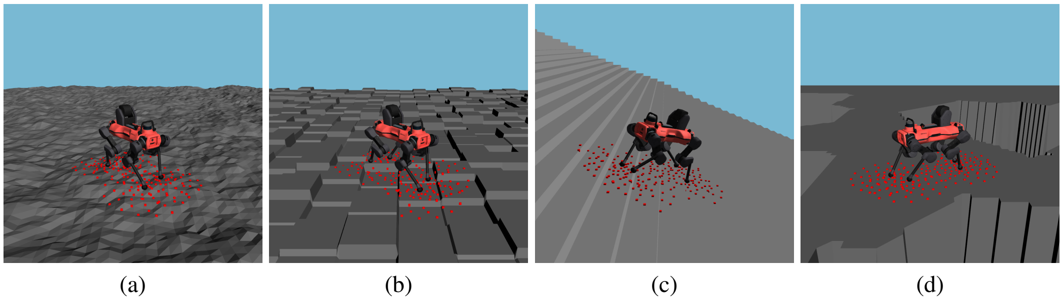

Figure 5: Examples of environments used for training of the locomotion policies. The difficulty of the terrain can be adjusted with parameters that modulate characteristics of the obstacles. The red dots illustrate the robot-centered height scan (exteroceptive observations) of the controller. The training environments can be composed of traversable (a, b, and c) or non-traversable (d) obstacles.

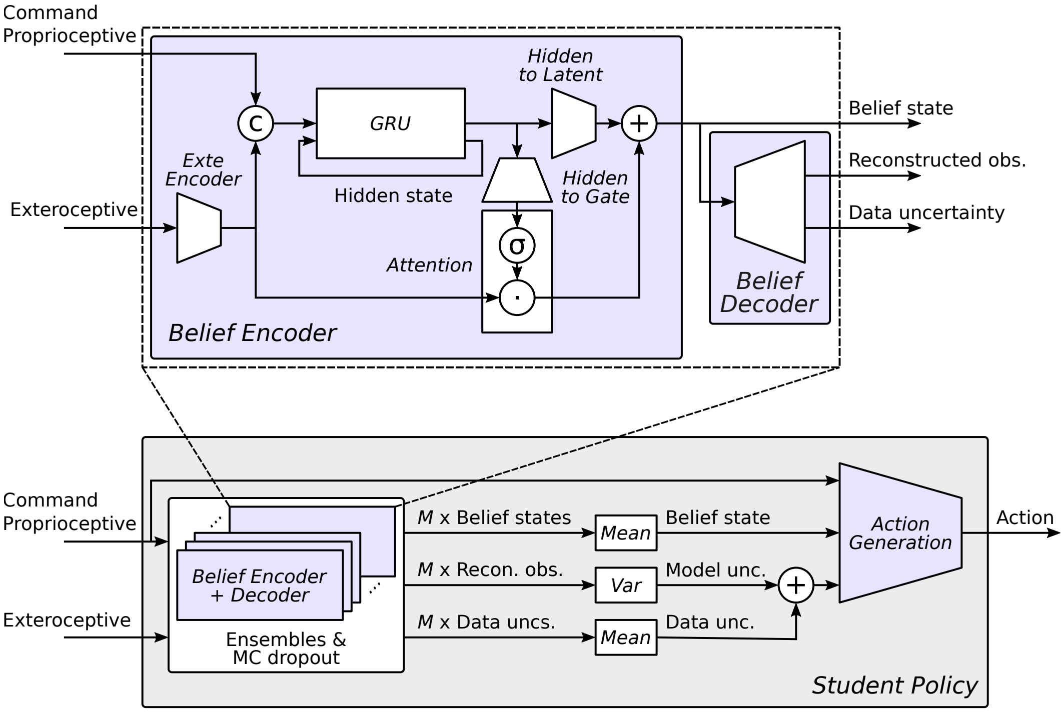

Figure 6: Details of the student architecture used during our experiments. The policy is composed of a set of belief encoder/decoder networks, which compute M estimates of the belief state, the reconstructed observations and data uncertainties. These M estimates are obtained from several MC dropout forward passes for each model in the ensemble. The model uncertainty is then computed from the variance in the reconstructed observations.

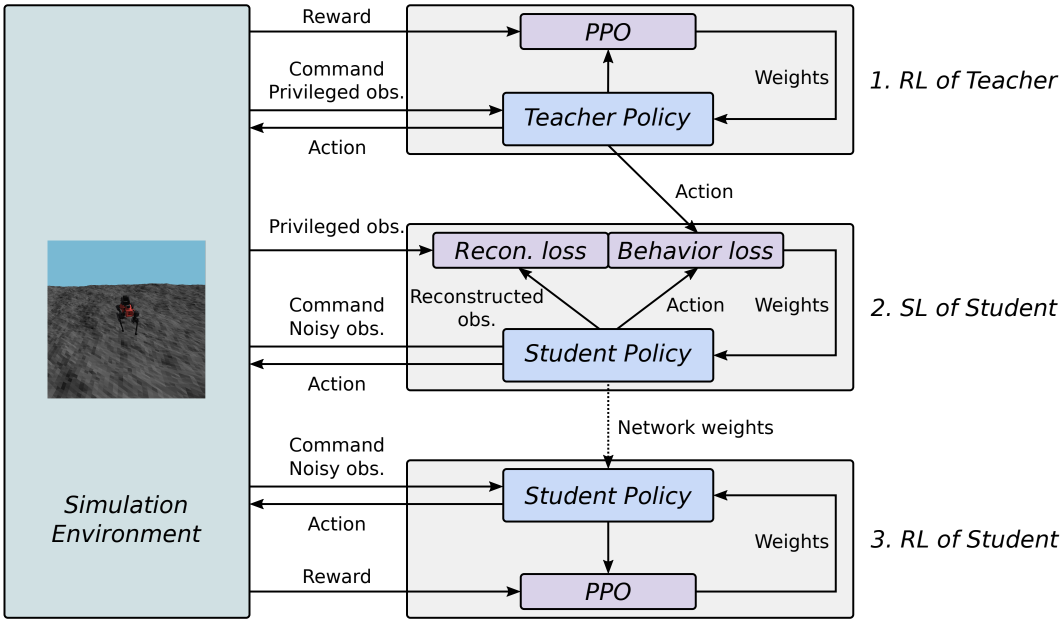

Figure 7: Steps used to train the final student policy.