REAL-TIME UNCERTAINTY DECOMPOSITION FOR ONLINE LEARNING CONTROL

Anonymous authors
Paper under double-blind review

ABSTRACT

Safety-critical decisions based on machine learning models require a clear understanding of the involved uncertainties to avoid hazardous or risky situations. While aleatoric uncertainty can be explicitly modeled given a parametric description, epistemic uncertainty rather describes the presence or absence of training data. This paper proposes a novel generic method for modeling epistemic uncertainty and shows its advantages over existing approaches for neural networks on various data sets. It can be directly combined with aleatoric uncertainty estimates and allows for prediction in real-time as the inference is sample-free. We exploit this property in a model-based quadcopter control setting and demonstrate how the controller benefits from a differentiation between aleatoric and epistemic uncertainty in online learning of thermal disturbances.

1 INTRODUCTION

With improved sensor quality and more powerful computational resources, data-driven models are increasingly applied in safety-critical domains such as autonomous driving or human-robot interaction (Grigorescu et al., 2020). However, measurements usually suffer from noise and the available data is often scarce compared to all possible states of a complex environment. This requires controllers, which rely on supervised learning techniques, to properly react to ignorance and imprecision in the model to avoid dangerous situations. In order to allow an implementation of risk-averse (for exploitation and safety improvements) or risk-seeking (for exploration) behavior, the model must clearly disaggregate the information in the data into more than just the "best estimate" and differentiate between different sources of uncertainty. Besides the point estimate of a model, one can distinguish aleatoric (uncertainty in the data) and epistemic (uncertainty in the model) uncertainty. The former is irreducible as it is inherent to the stochastic process the data is recorded from, while the latter origins from a limited expressive power of the model or scarce training samples (Der Kiureghian & Ditlevsen, 2009).

A promising approach in this field are Gaussian processes (GPs), which inherently provide a measure for its fidelity with the posterior variance prediction (Rasmussen & Williams, 2006). It also allows to differentiate aleatoric uncertainty (typically considered as observation noise) and epistemic uncertainty (modeled by the kernel). However, the former allows only homoscedastic (constant) estimates, while real-world applications typically require heteroscedastic uncertainty models. An extension to heteroscedastic GP regression is presented in (Lazaro-Gredilla & Titsias, 2011), however, it is a variational approximation and further increases the computational complexity of GPs, which is prohibitive when employing them for large data sets.

In deep learning, the modeling of uncertainties also gained increasing interest over the past years (Kendall & Gal, 2017). Heteroscedastic aleatoric uncertainty can be captured well, if the output of the stochastic process can directly be observed and its parametric distribution is known. However, for more general cases approximate inference techniques such as variational inference or sampling is required (Bishop, 2006). For epistemic uncertainty estimation with neural networks (NN), the key idea for most approaches can be summarized as follows. Randomness is introduced to the neural network through sampling during training and inference. While the training robustifies the network against the injected noise at the training locations, it allows the noise to pass to the output at input locations where no training data is available. For inference, multiple predictions of the network are sampled for the same inputs, allowing to compute a statistical measure for the

uncertainty at the output (Depeweg et al., 2018; Depeweg, 2019). However, sampling the network during inference is a high computational burden and is therefore not suitable in real-time critical control tasks.

Despite those drawbacks in the uncertainty representation of data-driven models, the control community started to incorporate them increasingly in the decision making for various applications. For example Fanger et al. (2016) uses an epistemic uncertainty measure to dynamically assign leader order follower roles for cooperative robotic manipulation. The work by Berkenkamp et al. (2016) ensures a safe exploration of an unknown task space based on GP error bounds and a gain scheduling approach for computed torque control was presented in Beckers et al. (2019). In Umlauft & Hirche (2020) and Chowdhary et al. (2015) an online learning control approach for GPs models is considered. More general, risk averse control strategies have been presented by Umlauft et al. (2018); Medina et al. (2013); Todorov & Li (2005). However, all of these approaches only consider the model fidelity and do not differentiate between aleatoric and epistemic uncertainty.

Differentiating between the different sources of uncertainty becomes crucial in online learning control, where a model-based policy simultaneously controls a system, while taking measurements from it and uses this data to improve its model Lee et al. (2017). As a poor data-driven model can lead to failure of the overall system, acquiring good training data is key to succeed in online learning task Lutter et al. (2019). For many application, the acquisition or measurements of new training is often costly, e.g. blood tests to track the severity of disease and hence, measurements should only be taken when necessary. Thus, without knowledge where the uncertainty originates from (missing or noisy data), the control cannot take proper decision for the exploitation-exploration trade-off.

The main contributions of this paper are the following. We propose a deep learning framework to allow a decomposition of aleatoric and epistemic uncertainty without sampling during inference. The resulting real-time capable online learning model is employed by a controller, which shows a distinct reaction to both types of uncertainties. We evaluate the proposed methods on synthetic and real-world benchmark data sets, and simulate a quadcopter learning online the disturbances injected by thermals.

2 Problem formulation

Consider the discrete-time dynamical system[1] with control $\boldsymbol{u} \in \mathbb{U} \subseteq \mathbb{R}^{d_u}$ and state $\boldsymbol{x} \in \mathbb{X} \subseteq \mathbb{R}^{d_x}$

$$\boldsymbol{x}_{k+1} = \boldsymbol{g}(\boldsymbol{x}_k, \boldsymbol{u}_k) + \boldsymbol{y}_k, \tag{1}$$

where $\boldsymbol{g} \colon \mathbb{X} \times \mathbb{U} \to \mathbb{X}$ is known, while $\boldsymbol{y}$ is a i.i.d. random vector sampled in every time step from

$$\boldsymbol{y}_k \sim \mathcal{D}(\boldsymbol{f}(\boldsymbol{x}_k)), \tag{2}$$

where $\mathcal{D}(\cdot)$ denotes a known distribution over real vectors $\boldsymbol{y} \in \mathbb{Y} \subseteq \mathbb{R}^{d_x}$ and depends on the parameters $\boldsymbol{p} \in \mathbb{P} \subseteq \mathbb{R}^{d_p}$. These state-dependent parameters arise from an unknown mapping $\boldsymbol{f} \colon \mathbb{X} \to \mathbb{P}$.

We denote the unknown component $\boldsymbol{y}_k$ of the dynamical system generally as disturbance but it could also be the unmodeled part of the dynamics, such as friction or serve as black-box model for the dynamics if no analytic description is available ($\boldsymbol{g}(\cdot, \cdot) = 0$). We assume measurements can be taken at every time step to obtain the data set $\mathbb{D}_{\text{tr}} = \{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^{N_{\text{tr}}}$ with inputs $\mathbb{X}_{\text{tr}} = \{\boldsymbol{x}_i\}_{i=1}^{N_{\text{tr}}}$ and outputs $\mathbb{Y}_{\text{tr}} = \{\boldsymbol{y}_i\}_{i=1}^{N_{\text{tr}}}$, such that a model $\hat{\boldsymbol{f}}(\cdot)$ of $\boldsymbol{f}(\cdot)$ can be learned. $N_{\text{tr}} \in \mathbb{N}$ denotes the current number of training data points and is initially zero, i.e., the training set is empty. We consider the measurements of $\boldsymbol{y}$ and their processing to be costly, such that we only want to add new training points when necessary in order to maintain a high data-efficiency.

Applications of such scenarios can be found in network or distributed systems, where multiple sensors share the same scarce communication channel, or autonomous and embedded systems with limited data storage capacity such as quadcopters, for instance. But also a medical doctor could be supported in the decision to demand for more costly analysis measurements to judge upon the health state of a patient. The scarce availability of data take and a high cost to collect new measurements makes a computationally efficient model fidelity estimate essential for online learning control systems

[1] Bold/capital symbols generally denote vectors/matrices, $\mathcal{D}(\cdot)$/$\mathcal{U}(\cdot)$/$\mathcal{N}(\cdot)$/$\mathcal{B}(\cdot)$ a general parametric/the uniform/Gaussian/Bernoulli distribution, respectively.

The goal is to control the system (1), such that it follows a reference $\boldsymbol{x}^{\text{des}}$. While classical control approaches allow to determine controls $\boldsymbol{u}$ using the distribution $\mathcal{D}(\boldsymbol{f}(\cdot))$, the data for learning the disturbance model $\hat{\boldsymbol{f}}(\cdot)$ must be selected online. Therefore, online learning control exhibits an additional design freedom, which cannot be neglected when high data-efficiency is of importance. In order to determine relevant training data, reliable epistemic uncertainty estimates must be computed in real-time. However, existing approaches for modeling epistemic uncertainty in deep learning suffer from a computational complexity which does not allow this. Therefore, we consider the problem of efficiently modeling epistemic and aleatoric uncertainty in neural networks, and investigate how they can be employed for data-efficient online learning control.

Before proposing a control strategy for this online learning problem, we first discuss possible deep learning techniques for epistemic uncertainty estimation and propose a novel method that suits the requirements for online learning. Epistemic uncertainty will be the key indicator whether new measurements should be taken or not.

3 Epistemic Uncertainty Estimation

3.1 Related Work

Learning an epistemic uncertainty estimator is not straight forward as it measures the absence of training data. Most prominently Gaussian processes with stationary kernels offer such a measure implicitly with their posterior variance prediction. However, GPs are known to scale poorly for large data sets: While regression and uncertainty predictions can be preformed with $\mathcal{O}(N_{\text{tr}})$ and $\mathcal{O}({N_{\text{tr}}}^2)$, respectively, considering a new data point takes $\mathcal{O}({N_{\text{tr}}}^3)$ computations (also without hyperparameter optimization). While various methods have been proposed to make GP computationally more efficient, including sparse GPs (Quinonero-Candela & Rasmussen, 2005), distributed GPs (Deisenroth & Ng, 2015) and local GPs (Nguyen-Tuong et al., 2009a;b), these approximations typically focus only on the precision of the point estimate and distort the uncertainty prediction.

More recently, several different approaches for epistemic uncertainty estimates using deep learning frameworks have been proposed. Popular approaches rely on Bayesian approximations(Depeweg et al., 2016) or permanent dropouts (not only during training to avoid overfitting) (Gal, 2016; Gal & Ghahramani, 2016). Furthermore, latent inputs can also be used to achieve a decomposition into aleatoric and epistemic uncertainty as presented in (Depeweg et al., 2017). However, in particular for Bayesian NNs, these approaches become computationally challenging. Firstly, they have a larger number of parameters to tune than their deterministic counterparts and rely on variational inference methods (Kwon et al., 2020). Secondly, the prediction requires to sample the entire network before the statistics of the output can be computed. For the application in real-time critical control problems (e.g., robotics with a sampling rate of $1\,\text{kHz}$), these computational burdens prohibit an employment of these techniques. A sampling-free estimation methods is proposed by Postels et al. (2019), but suffers from a quadratic space complexity in the number of weights in the network and relies on first-order Taylor approximations in the propagation of the uncertainties, which might become inaccurate depending on the non-linearity of the activation functions

3.2 *EpiOut*- explicitly learning epistemic uncertainty

In order to allow the estimation of epistemic uncertainty in real-time, we introduce the idea of explicitly modeling it with a separate output of a neural network, calling it *EpiOut*. Since the epistemic uncertainty expresses the absence of data, the original data set $\mathbb{D}_{\text{tr}}$ does not contain data for training *EpiOut*. Therefore, we generate an epistemic uncertainty data set, with inputs $\mathbb{X}_{\text{epi}} = \{\tilde{\boldsymbol{x}}_j\}_{j=1}^{N_{\text{epi}}}$ and outputs $\mathbb{Y}_{\text{epi}} = \{\tilde{y}_j\}_{j=1}^{N_{\text{epi}}}$ concatenated in $\mathbb{D}_{\text{epi}} = \{(\tilde{\boldsymbol{x}}_j, \tilde{y}_j)\}_{j=1}^{N_{\text{epi}}}$, $N_{\text{epi}} \in \mathbb{N}$.

Different variations for sampling the set $\mathbb{X}_{\text{epi}}$ can be chosen depending on the desired scalability properties. A naive approach is to sample the entire input space uniformly, which suffers from the curse of dimensionality. Alternatively, we propose to sample around existing training points from a

normal distribution

$$\mathbb{X}_{\text{epi}} = \bigcup_{i=1}^{N_{\text{tr}}} \{\tilde{\boldsymbol{x}}_j \sim \mathcal{N}(\boldsymbol{x}_i, \Gamma), j = 1, \ldots, N_{\text{epi}}/N_{\text{tr}}\}, \tag{3}$$

where we implicitly assume that N_{epi} is chosen such that $\delta = N_{\text{epi}}/N_{\text{tr}}$ is a positive integer. The scaling $\Gamma = \text{diag}(\gamma_1, \ldots, \gamma_{d_x})$, with $\gamma_i \in \mathbb{R}_+$ can be interpreted similarly to the lengthscale of a GP as a measure for how far away from a training point the prediction is reliable.

We propose to set δ to a multiple of $2d_x + 1$ which corresponds to the intuition to pad each training point in both directions of each dimension with a *epi point* $\tilde{\boldsymbol{x}}$. The reasoning behind the additional $+1$ point will become clear in the following. To define the set $\mathbb{Y}_{\text{epi}}$, we first compute the minimal distance (according to any distance metric $d\colon \mathbb{X} \times \mathbb{X} \to \mathbb{R}_{0,+}$) to the training data for each epi point

$$d_j = \min_{\boldsymbol{x} \in \mathbb{X}_{\text{tr}}} d(\tilde{\boldsymbol{x}}_j, \boldsymbol{x}), \quad j = 1, \ldots, N_{\text{epi}}, \tag{4}$$

keeping in mind that the closest training data point is not necessarily the one used to generate the sample. Let $d_{N_{\text{tr}}}$ be the N_{tr}-th smallest element of all d_j, we generate $\mathbb{Y}_{\text{epi}}$ and update $\mathbb{X}_{\text{epi}}$ as follows

$$\tilde{y}_j = \begin{cases} 1, & \text{if } d_j > d_{N_{\text{tr}}} \\ 0, \quad \tilde{\boldsymbol{x}}_j \leftarrow \arg\min_{\boldsymbol{x} \in \mathbb{X}_{\text{tr}}} d(\tilde{\boldsymbol{x}}_j, \boldsymbol{x}) & \text{if } d_j \leq d_{N_{\text{tr}}} \end{cases}. \tag{5}$$

Thus, the N_{tr} points in $\mathbb{X}_{\text{epi}}$ with the least distance to a training point are replaced by the corresponding point in $\mathbb{X}_{\text{tr}}$. Now the choice of $2d_x + 1$ epi points becomes clear as one of them will be turned into $\tilde{y} = 0$ corresponding to "low epistemic uncertainty", while $2d_x$ points further away from the training point $\tilde{y} = 0$ indicate the opposite.

Given the data set $\mathbb{D}_{\text{epi}}$, the neural network is now equipped with one additional output, i.e., the parameter layer is $d_p + 1$ dimensional with output $[\hat{\boldsymbol{f}}(\cdot)\ \eta(\cdot)]^T$. The new output $\eta(\cdot)$ is terminated with a neuron using a sigmoidal activation function, such that $\eta\colon \mathbb{X} \to [0, 1]$. This is beneficial because it allows immediately to judge, whether the predicted uncertainty is *high* (≈ 1) or *low* (≈ 0) without any reference evaluation (see comparison to alteranative methods below).

Independently of the loss function for the original network, the augmented output, also considered as *epistemic output* is trained using a binary cross-entropy loss, which is the natural choice for binary outputs. It quantifies the uncertainty in the prediction of the other outputs based on the distance to the training data measured by $d(\cdot, \cdot)$. For the sake of focus, we will be using the Euclidean distance, however the method can be easily extended to other metrics and we leave it to future work to investigate alternatives.

3.3 Computational complexity

The analysis of the computational complexity shows that equation 3 is a $\mathcal{O}(N_{\text{epi}}) \widehat{=} \mathcal{O}(N_{\text{tr}} d_x)$ operation, whereas equation 4 is for a trivial implementation a $\mathcal{O}(N_{\text{tr}} N_{\text{epi}}) \widehat{=} \mathcal{O}(d_x {N_{\text{tr}}}^2)$ operation. However, an implementation based on kd-tree (Cormen et al., 2009) allows an execution in $\mathcal{O}(N_{\text{epi}} \log(N_{\text{epi}})) \widehat{=} \mathcal{O}(d_x N_{\text{tr}} \log(N_{\text{tr}} d_x)))$ time. Finding the N_{tr} smallest distances from all N_{epi} points in equation 5 can obtained in $\mathcal{O}(N_{\text{tr}} + (N_{\text{epi}} - N_{\text{tr}}) \log(N_{\text{tr}})) \widehat{=} \mathcal{O}(N_{\text{tr}} + N_{\text{tr}}(d_x - 1) \log(Ntr))$ time. The training of neural network with a fixed number of weights requires $\mathcal{O}(N_{\text{epi}}) \widehat{=} \mathcal{O}(N_{\text{tr}} d_x)$. Hence, the overall complexity results in $\mathcal{O}(d_x N_{\text{tr}} \log(d_x N_{\text{tr}}))$, and it is straight forward to derive an overall space complexity of $\mathcal{O}(N_{\text{epi}} d_x) \widehat{=} \mathcal{O}(N_{\text{tr}} {d_x}^2)$ for storing the set $\mathbb{X}_{\text{epi}}$. The following should be considered when comparing to classical deep learning frameworks which generally can be trained in linear time.

- When used on streaming data (as for online learning control), the set $\mathbb{D}_{\text{epi}}$ can be constructed iteratively, reducing the complexity to $\mathcal{O}(\log(N_{\text{tr}}))$
- The most time critical computation equation 4 can efficiently be parallelized on a GPU.
- The method is designed for applications where measuring data is considered costly and therefore sparse data can be expected.

3.4 Evaluation

For evaluation we compare the following models. (Implementation in the supplementary material)

- vanilla *GP model* with a squared exponential automatic relevance determination kernel based on the GPy implementation[2]
- *BNN*: Bayesian Neural Network with 2 fully connected hidden layers each with 50 hidden units and normal distributions over their weights based on this implementation.[3]
- *Dropout*: A neural network with 2 fully connected permanent layers each with 50 hidden units with dropout probability $\rho = 0.05$. [4].
- *EpiOut*: The proposed model with 2 fully connected layers (50 neurons) and $\Gamma = \boldsymbol{I}$, $\delta = 2$.

For the evaluation we utilize a weighted mean square error measures defined as follows

$$\rho = \frac{\sum_{i=1}^{N_{\text{te}}} (y_i - \hat{f}(\boldsymbol{x}_i))^2 (1 - \eta(\boldsymbol{x}_i))}{\sum_{i=1}^{N_{\text{te}}} (1 - \eta(\boldsymbol{x}_i))}, \tag{6}$$

i.e., if the model decides that it is uncertain about the prediction at a test point, the squared error for this prediction is discounted (weighted less). However, the model can only achieve a low ρ if it is also certain at some test points, because the denominator, goes to zero for many uncertain predictions. In consequence, ρ is only defined if $\eta(\cdot) < 1$ holds for at least one test point. Furthermore, the total discount, defined as $\sum_{i=1}^{N} \eta(\boldsymbol{x}_i)$ can additionally be utilized for a plausibility check of the epistemic uncertainty predictions since it should generally be larger on the test than on the training data set.

The measure in equation 6 relies on epistemic uncertainty prediction in the interval $[0, 1]$. This is only ensured for the proposed *EpiOut*approach and therefore the uncertainty measures provided by the GP, *Dropout* and *BNN* are scaled to the unit interval based on the evaluation on all test points.

The following data sets are utilized for evaluation.

- *1D Center*: The nominal function is $f(x) = \sin(x\pi)$, with training points $\mathbb{X}_{\text{tr}} = \{x_i \sim \mathcal{U}(-1, 1)\}_{i=1}^{100}$ and $N_{\text{te}} = 961$ test points are placed on a grid $[-4, 4]$.
- *1D Split*: Same as *1D Center*, but $\mathbb{X}_{\text{tr}} = \{x_i \sim \mathcal{U}(-2, -1)\}_{i=1}^{100} \cup \{x_i \sim \mathcal{U}(1, 2)\}_{i=101}^{200}$.
- *2D Gaussian*: The nominal function ($d_x = 2$,$d_p = 1$) is $f(\boldsymbol{x}) = \frac{\sin(5x_1)}{5x_1} + x_2^2$ with training points $\mathbb{X}_{\text{tr}} = \left\{\boldsymbol{x}_i \sim \mathcal{N}\left(\begin{bmatrix}-1\\0\end{bmatrix}, \begin{bmatrix}0.02 & 0\\0 & 0.1\end{bmatrix}\right)\right\}_{i=1}^{500} \cup \left\{\boldsymbol{x}_i \sim \mathcal{N}\left(\begin{bmatrix}1\\0\end{bmatrix}, \begin{bmatrix}0.02 & 0\\0 & 0.1\end{bmatrix}\right)\right\}_{i=501}^{1000}$ and $N_{\text{te}} = 961$ test points are uniformly placed on a grid $[-2, 2]^2$.
- *2D Square*: Same as *2D Gaussian*, but with with $N_{\text{tr}} = 80$ training points placed uniformly along the boundary of the square $[-1, 1]^2$.
- *PMSM temperature* is a 2Hz recording ($d_x = 8$, $d_y = 1$) of the temperature from a permanent magnet synchronous motor.[5]. To allow a comparison with the GP within reasonable computational limits, $N_{\text{tr}} = 5000$ and $N_{\text{te}} = 1000$ points were randomly extracted from a total of $\approx 10^6$ samples.
- *Sarcos* is a data set for learning the inverse dynamics of a seven degrees-of-freedom SARCOS anthropomorphic robot arm $d_x = 21$, $d_p = 1$.[6]. $N_{\text{tr}} = 10000$ and $N_{\text{te}} = 2000$ points were randomly extracted from a total of $\approx 5 \times 10^4$ samples.

3.5 Results & Discussion

The numerical results are presented in Table 1 and an illustration for the data set *1D Split* for all models is shown in Fig. 1. Besides showing empirically an advantage over existing approaches we want to point out the following benefits.

[2] https://sheffieldml.github.io/GPy/
[3] https://matthewmcateer.me/blog/a-quick-intro-to-bayesian-neural-networks/
[4] https://github.com/yaringal/DropoutUncertaintyDemos
[5] https://www.kaggle.com/wkirgsn/electric-motor-temperature
[6] http://www.gaussianprocess.org/gpml/data/

Table 1: Weighted mean squared error ρ as defined in equation 6 for the considered models on different data sets. The GP model is grayed out since it does not scale towards larger data sets.

	1D center	1D split	2D Gaussian	2D square	Motor temperature	Sarcos
GPmodel	0.245159	0.182383	0.075812	0.033676	0.001075	4.96823
BNN	1.848515	0.357473	1.164467	1.145277	0.127809	21.46700
Dropout	0.689114	0.617375	1.105933	0.381499	0.096435	24.04090
EpiOut	**0.012303**	**0.009561**	**0.063902**	**0.040084**	**0.005703**	**16.33120**

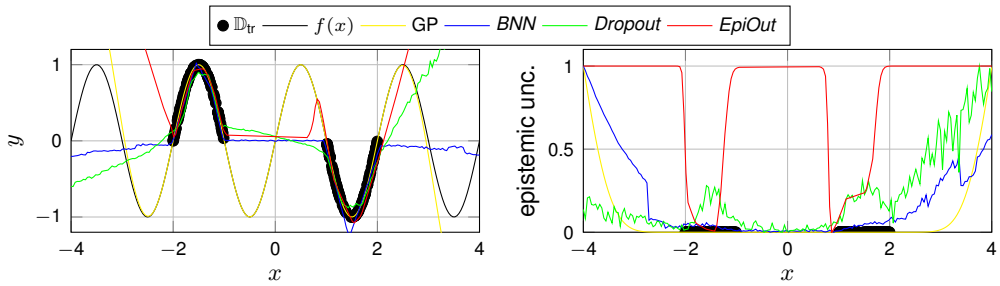

Figure 1: The point estimate (mean prediction) for the different models along with the training data *1D split* and the true underlying function $f(x) = \sin(\pi x)$ are shown on the left. The right plot shows the epistemic uncertainty estimate, where *BNN* and *Dropout* clearly miss to predict a higher uncertainty between the data clusters.

- The *EpiOut*model predicts the uncertainty measure in a sample free manner. This is crucial in data-efficient online learning scenarios, where the epistemic uncertainty is used to evaluate the usefulness of an incoming data point to decide upon its rejection. Hence, it is called more frequently than the online training function and must be computationally efficient. The prediction time for *EpiOut*is typically an order of magnitude faster than *Dropout* and *BNN* (For the exact numbers, refer to the supplementary material).
- A single evaluation of $\eta(\cdot)$ is sufficient for a conclusion whether the uncertainty is high or low, since it is bounded to the interval $[0, 1]$, whereas alternative approaches based *BNN* and *Dropout* provide a return value $[0, \infty]$, which can be difficult to interpret without a maximum value as reference.
- For many data sets, the existing methods (*Dropout*, *BNN*) have a larger total discount on the training set than on the test set, which does not correspond to the expectation for en epistemic uncertainty estimate, see the supplementary material for the exact numbers.

4 Online Learning Control using Uncertainty Distinction

4.1 Learning control design

Since the computation of epistemic uncertainty estimates using the *EpiOut*approach is significantly faster compared to existing methods, it can be directly applied within real-time control loops. This allows us to continuously monitor the epistemic uncertainty in order to decide when additional training data is necessary. While it would be straightforward to set a threshold on the epistemic uncertainty for determining necessary measurements $\boldsymbol{y}$, this approach requires the tuning of an additional parameter and is potentially prone to regression errors of *EpiOut*$\eta(\cdot)$. Therefore, we propose to add measurements randomly to the training data set by sampling from a Bernoulli distribution, whose probability parameter corresponds to $\eta(\cdot)$. Hence, given a new measurement $(\boldsymbol{x}_i, \boldsymbol{y}_i)$, the training data set is updated according to

$$\mathbb{D}_{\text{tr}} \leftarrow \mathbb{D}_{\text{tr}} \cup \begin{cases} (\boldsymbol{x}_i, \boldsymbol{y}_i) & \text{if i} = 1 \\ \emptyset & \text{if i} = 0 \end{cases} \qquad \text{where i} \sim \mathcal{B}(\alpha), \alpha = \eta(\boldsymbol{x}_i). \tag{7}$$

The measurement strategy (7) ensures a high accuracy of the disturbance model $\hat{\boldsymbol{f}}(\cdot)$ by guaranteeing a training data set which contains samples in the proximity of the current state $\boldsymbol{x}_i$. However, the system (1) is inherently random due to the stochastic nature of the disturbance $\boldsymbol{f}(\cdot)$. Therefore, we employ a feedback control law

$$\boldsymbol{u} = \boldsymbol{K}\left(\boldsymbol{x} - \boldsymbol{x}^{\text{des}}\right) + \boldsymbol{u}_{\text{ff}}, \tag{8}$$

where $\boldsymbol{u}_{\text{ff}}$ is a feedforward control term determined based on the known model $\boldsymbol{g}(\cdot,\cdot)$ (e.g. the gravitational force on the quadcopter) and the learned disturbance model $\hat{\boldsymbol{f}}(\cdot)$. To compensate for error is the latter model and the stochasticity of the disturbance, additionally use a linear feedback with gain matrix $\boldsymbol{K} \in \mathbb{R}^{d_x \times d_x}$, which we consider to be sufficient to achieve a stable closed-loop.

Generally high control gains lead to high tracking performance, but are usually avoided because they lead to a higher energy consumption for the control effort and reinforce measurement noise, which can lead to instability. Therefore, it is generally advisable to let the feedforward term $\boldsymbol{u}_{\text{ff}}$ take over most of the control effort and keep the feedback term small whenever possible. Therefore, we adapt gains to the aleatoric uncertainty expressed by $\mathcal{D}(\hat{\boldsymbol{f}}(\boldsymbol{x}))$, i.e.,

$$\boldsymbol{K} = \bar{k}\big(\boldsymbol{I} + \beta \text{diag}(\text{Var}(\boldsymbol{y}))\big), \tag{9}$$

where $\boldsymbol{y} \sim \mathcal{D}(\hat{\boldsymbol{f}}(\boldsymbol{x}))$, $\beta \in \mathbb{R}_{0,+}$ is the sensitivity, $\bar{k} \in \mathbb{R}_+$ defines the minimum control gain, $\text{diag}(\cdot)$ returns a diagonal matrix and $\text{Var}(\cdot)$ denotes the variance operator. This allows us to robustify the closed-loop against the process noise and can even guarantee stability Beckers et al. (2019), while at the same time we can keep the energy consumption low. Note that our approach differs from existing methods for uncertainty-based gain scheduling such as, e.g., Fanger et al. (2016); Beckers et al. (2019). We employ only aleatoric uncertainty to tune the feedback gains, since it cannot be reduced through additional training data. In contrast, existing methods include epistemic uncertainty, which can be more efficiently dealt with using online learning.

4.2 Evaluation in a Quadcopter Control Task

To demonstrate the application relevance of our proposed approach, we consider the task of a quadcopter to explore a terrain with unknown thermals. We assume that the quadcopter dynamics are known and set the function $g(\cdot)$ accordingly. The thermals act on the quadcopter as a disturbance only in the z-direction, such that $\boldsymbol{y}$ can be different from 0 only in the entries corresponding to the z-direction. We model the disturbance as normal distribution $\mathcal{N}(\mu(\boldsymbol{x}), \sigma^2(\boldsymbol{x}))$, leading to $\boldsymbol{f}(\boldsymbol{x}) = [\mu(\boldsymbol{x}), \sigma(\boldsymbol{x})]^T$ as distribution parameters of the aleatoric uncertainty which have to be estimated by the first d_p outputs of the NN. The data of the thermals is taken from publicly available paragliding data (`https://thermal.kk7.ch`). We control the quadcopter using the proposed online learning control method with aleatoric uncertainty dependent feedback gains (9) and the measuring strategy (7). Due to the fact that merely the z-direction exhibits aleatoric uncertainty, the feedback gains are only adapted for the corresponding states. The desired trajectory $\boldsymbol{x}^{\text{des}}$ is a square in the x^q-y^q-plane with edge length 0.1 and constant height $z^q = 0$ and $\beta = 2$.

4.3 Simulation Results

The tracking performance of the quadcopter model is illustrated in Fig. 2. The disturbance model after completing three cycles is shown in Fig. 4. Our approach significantly reduces the tracking error while yielding a high data-efficiency.(Implementation in the supplementary material) Figure 3 shows how that after the first round most of the required data is already recorded.

5 Conclusion

This paper presents a novel deep learning structure for decomposing epistemic and aleatoric uncertainty and proposes an advanced control framework making distinct use of these uncertainty measures. As the prediction by the model are obtained sample-free, it allows for real-time critical online learning and outperforms existing methods on the proposed uncertainty-weighted precision measure. The proposed online learning control algorithm is inherently data-efficient by adding only required points to the data set. For future work will move towards a model, which forgets about incorporated data points to allow a life-long learning.

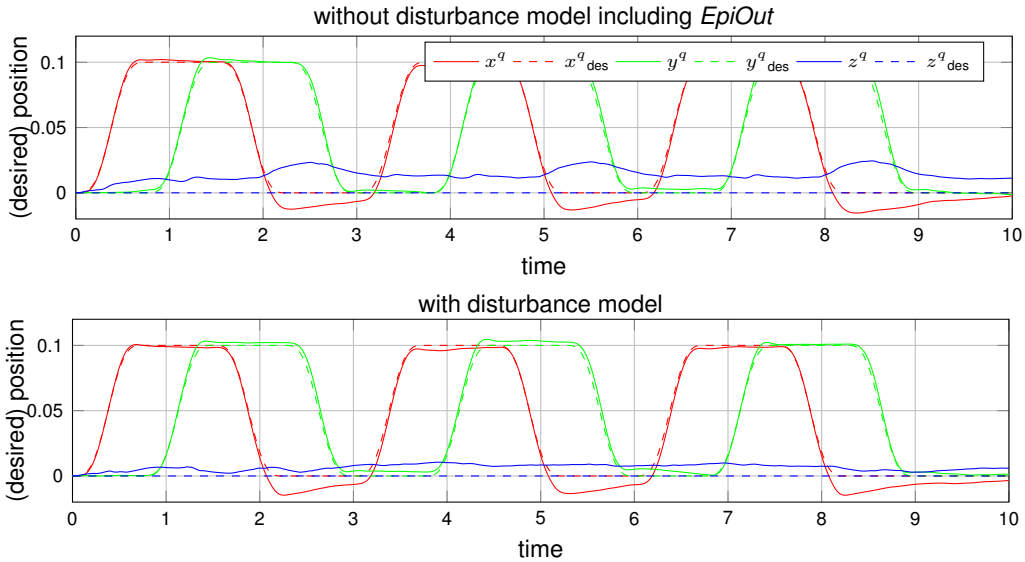

Figure 2: The tracking performance of the quadcopter in z direction, (where the disturbance acts) is significantly improved (top without vs bottom with) a disturbance estimation model.

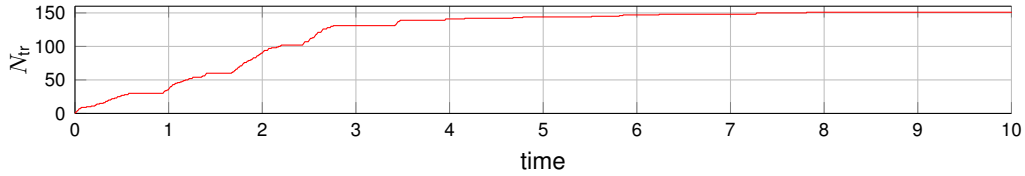

Figure 3: Number of training points over time. The decision to add a new training point depends on the epistemic uncertainty estimate.

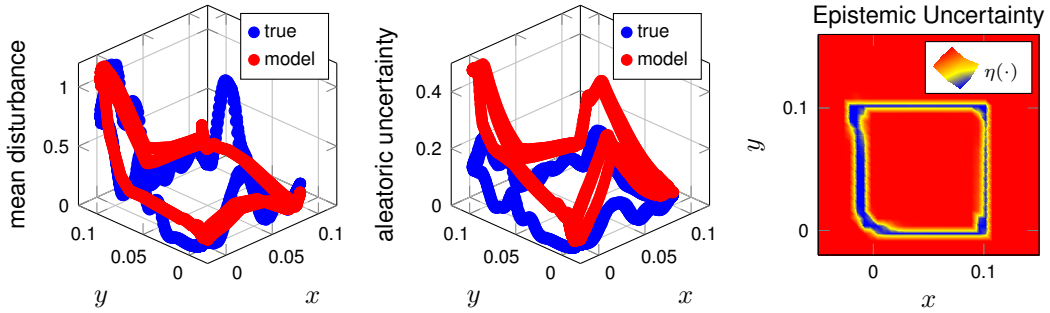

Figure 4: The mean disturbance model (left) captures most of the thermals, and the aleatoric uncertainty (middle) is slightly overestimated by the model. The "true" values are obtained from 1000 samples of the disturbance. The epistemic uncertainty (right) shows that the model is only confident close to the desired trajectory.

REFERENCES

Thomas Beckers, Dana Kulić, and Sandra Hirche. Stable Gaussian process based tracking control of Euler-Lagrange systems. *Automatica*, 23(103):390–397, 2019. doi: 10.1016/j.automatica.2019.01.023.

Felix Berkenkamp, Riccardo Moriconi, Angela Schoellig, and Andreas Krause. Safe learning of regions of attraction for uncertain, nonlinear systems with Gaussian processes. *arXiv preprint arXiv:1603.04915*, 2016.

Christopher M Bishop. *Pattern recognition and machine learning*. springer, 2006.

Girish Chowdhary, Hassan A. Kingravi, Jonathan. P. How, and Patricio A. Vela. Bayesian nonparametric adaptive control using Gaussian processes. *IEEE Transactions on Neural Networks and Learning Systems*, 26(3):537–550, March 2015. ISSN 2162-237X. doi: 10.1109/TNNLS.2014.2319052.

Thomas H Cormen, Charles E Leiserson, Ronald L Rivest, and Clifford Stein. *Introduction to algorithms*. MIT press, 2009.

Marc Deisenroth and Jun Wei Ng. Distributed Gaussian processes. In *International Conference on Machine Learning (ICML)*, pp. 1481–1490, 2015.

Stefan Depeweg. *Modeling Epistemic and Aleatoric Uncertainty with Bayesian Neural Networks and Latent Variables*. PhD thesis, Technische Universität München, 2019.

Stefan Depeweg, José Miguel Hernández-Lobato, Finale Doshi-Velez, and Steffen Udluft. Learning and policy search in stochastic dynamical systems with Bayesian neural networks. *arXiv preprint arXiv:1605.07127*, 2016.

Stefan Depeweg, José Miguel Hernández-Lobato, Finale Doshi-Velez, and Steffen Udluft. Uncertainty decomposition in Bayesian neural networks with latent variables. *arXiv preprint arXiv:1706.08495*, 2017.

Stefan Depeweg, Jose-Miguel Hernandez-Lobato, Finale Doshi-Velez, and Steffen Udluft. Decomposition of uncertainty in bayesian deep learning for efficient and risk-sensitive learning. In *International Conference on Machine Learning*, pp. 1184–1193. PMLR, 2018.

Armen Der Kiureghian and Ove Ditlevsen. Aleatory or epistemic? does it matter? *Structural safety*, 31(2):105–112, 2009.

Yunis Fanger, Jonas Umlauft, and Sandra Hirche. Gaussian processes for dynamic movement primitives with application in knowledge-based cooperation. In *International Conference on Intelligent Robots and Systems (IROS)*, pp. 3913–3919. IEEE, October 2016. doi: 10.1109/IROS.2016.7759576.

Yarin Gal. *Uncertainty in Deep Learning*. PhD thesis, University of Cambridge, September 2016.

Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pp. 1050–1059, New York, New York, USA, June 2016. PMLR.

Sorin Grigorescu, Bogdan Trasnea, Tiberiu Cocias, and Gigel Macesanu. A survey of deep learning techniques for autonomous driving. *Journal of Field Robotics*, 37(3):362–386, 2020. doi: 10.1002/rob.21918.

Alex Kendall and Yarin Gal. What uncertainties do we need in Bayesian deep learning for computer vision? In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 5574–5584. Curran Associates, Inc., 2017.

Yongchan Kwon, Joong-Ho Won, Beom Joon Kim, and Myunghee Cho Paik. Uncertainty quantification using bayesian neural networks in classification: Application to biomedical image segmentation. *Computational Statistics and Data Analysis*, 142:106816, 2020. ISSN 0167-9473. doi: 10.1016/j.csda.2019.106816.

Miguel Lazaro-Gredilla and Michalis Titsias. Variational heteroscedastic gaussian process regression. In *International Conference on Machine Learning (ICML)*, pp. 841–848, New York, NY, USA, June 2011. ACM. ISBN 978-1-4503-0619-5.

Kit-Hang Lee, Denny K.C. Fu, Martin C.W. Leong, Marco Chow, Hing-Choi Fu, Kaspar Althoefer, Kam Yim Sze, Chung-Kwong Yeung, and Ka-Wai Kwok. Nonparametric online learning control for soft continuum robot: An enabling technique for effective endoscopic navigation. *Soft Robotics*, 4(4):324–337, 2017. doi: 10.1089/soro.2016.0065. PMID: 29251567.

Michael Lutter, Christian Ritter, and Jan Peters. Deep lagrangian networks: Using physics as model prior for deep learning. In *International Conference on Learning Representations (ICLR)*, 2019.

Jose R. Medina, Dominik Sieber, and Sandra Hirche. Risk-sensitive interaction control in uncertain manipulation tasks. In *International Conference on Robotics and Automation (ICRA)*, pp. 502–507. IEEE, May 2013. doi: 10.1109/ICRA.2013.6630621.

Duy Nguyen-Tuong, Jan R. Peters, and Matthias Seeger. Local Gaussian process regression for real time online model learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 1193–1200. Curran Associates, Inc., 2009a.

Duy Nguyen-Tuong, Matthias Seeger, and Jan Peters. Model learning with local Gaussian process regression. *Advanced Robotics*, 23(15):2015–2034, 2009b.

Janis Postels, Francesco Ferroni, Huseyin Coskun, Nassir Navab, and Federico Tombari. Sampling-free epistemic uncertainty estimation using approximated variance propagation. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2931–2940, 2019.

Joaquin Quinonero-Candela and Carl E. Rasmussen. A unifying view of sparse approximate Gaussian process regression. *The Journal of Machine Learning Research*, 6:1939–1959, 2005.

Carl E. Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning*. MIT Press, Cambridge, MA, USA, January 2006.

Emanuel Todorov and Weiwei Li. A generalized iterative LQG method for locally-optimal feedback control of constrained nonlinear stochastic systems. In *American Control Conference (ACC)*, pp. 300–306. IEEE, 2005.

Jonas Umlauft and Sandra Hirche. Feedback linearization based on Gaussian processes with event-triggered online learning. *IEEE Transactions on Automatic Control (TAC)*, pp. 1–16, 2020. ISSN 1558-2523. doi: 10.1109/TAC.2019.2958840.

Jonas Umlauft, Lukas Pöhler, and Sandra Hirche. An uncertainty-based control Lyapunov approach for control-affine systems modeled by Gaussian process. *IEEE Control Systems Letters*, 2(3): 483–488, July 2018. ISSN 2475-1456. doi: 10.1109/LCSYS.2018.2841961.