

Deep Learning based Uncertainty Decomposition for Real-time Control

Neha Das[†]* Jonas Umlauf[†]* Armin Lederer*
Alexandre Capone* Thomas Beckers** Sandra Hirche*

* *School of Computation, Information and Technology, Technical University of Munich, Munich, Germany (e-mail: {neha.das, jonas.umlauft, armin.lederer, alexandre.capone, hirche}@tum.de)*

** *Department of Computer Science, Vanderbilt University, Nashville, TN 37212, USA (e-mail: thomas.beckers@vanderbilt.edu)*

Abstract: Data-driven control in unknown environments requires a clear understanding of the involved uncertainties for ensuring safety and efficient exploration. While aleatoric uncertainty that arises from measurement noise can often be explicitly modeled given a parametric description, it can be harder to model epistemic uncertainty, which describes the presence or absence of training data. The latter can be particularly useful for implementing exploratory control strategies when system dynamics are unknown. We propose a novel method for detecting the absence of training data using deep learning, which gives a continuous valued scalar output between 0 (indicating low uncertainty) and 1 (indicating high uncertainty). We utilize this detector as a proxy for epistemic uncertainty and show its advantages over existing approaches on synthetic and real-world datasets. Our approach can be directly combined with aleatoric uncertainty estimates and allows for uncertainty estimation in real-time as the inference is sample-free unlike existing approaches for uncertainty modeling. We further demonstrate the practicality of this uncertainty estimate in deploying online data-efficient control on a simulated quadcopter acted upon by an unknown disturbance model.

Keywords: Machine learning, learning for control, uncertain systems, real-time uncertainty estimation, data-efficient control

1. INTRODUCTION

Data-driven approaches are increasingly applied for system identification in safety-critical domains such as autonomous driving or human-robot interaction (Grigorescu et al., 2020). These techniques typically infer the unknown system dynamics from a dataset of measurements and use the inferred dynamics model for control. However, these datasets often suffer from noise and sparsity which can lead to modeling errors. Many control strategies have benefited from estimating the model’s fidelity in addition to the point estimate of the system outputs. In particular, estimates of the model’s fidelity have been incorporated in risk-averse control strategies to mitigate the risks induced by modeling errors (Umlauf et al., 2018), as well as for safe exploration (Liu et al., 2020a), event-triggered learn-

ing (Umlauf and Hirche, 2020), data-efficient exploration (Capone et al., 2020) among others.

Many of these approaches differentiate between the exploitation of the two different sources of the uncertainty - measurement noise, which is irreducible, also commonly referred to as aleatoric uncertainty, and epistemic uncertainty, that arises from data scarcity and can be combated by incorporating more measurements (Gal, 2016). For example, epistemic uncertainty is utilized by Umlauf and Hirche (2020) for event-triggered learning and Depeweg et al. (2018) lays down a framework for risk-sensitive online learning which can explicitly balance between the aleatoric and epistemic uncertainty components. Implementation of such approaches necessitates the decomposition of epistemic and aleatoric uncertainty in real-time. In particular, our interest lies in the implementation of a uncertainty-based event-triggered adaptive control approach for mobile platforms with large (nearly 1 kHz) sampling rates. The underlying idea is that with real-time uncertainty decomposition, epistemic uncertainty estimates can be obtained at or close to the system’s sampling rate and measurements taken for learning the dynamics model can be limited to regions with high epistemic uncertainty, thus increasing data-efficiency while at the same time providing a means to adapt control for better tracking performance.

The field of uncertainty-estimation encompasses a wide-variety of approaches. Gaussian processes (GPs) inher-

* This work was supported by the European Research Council (ERC) Consolidator Grant "Safe data-driven control for human-centric systems (CO-MAN)" under grant agreement number 864686, and TUM AGENDA 2030, funded by the Federal Ministry of Education and Research (BMBF) and the Free State of Bavaria under the Excellence Strategy of the Federal Government and the Länder as well as by the Hightech Agenda Bavaria.

**© 2023 Neha Das, Jonas Umlauf, Armin Lederer, Alexandre Capone, Thomas Beckers, Sandra Hirche. This work has been accepted to IFAC for publication under a Creative Commons Licence CC-BY-NC-ND"

† Both authors contributed equally.

ently provide a measure for their own fidelity with posterior standard deviation (Rasmussen and Williams, 2006) and can differentiate between the two kinds of uncertainties but generally suffer from poor scaling to large datasets (Quinero-Candela and Rasmussen, 2005). While various methods have been proposed to make GP computationally more efficient (Snelson and Ghahramani, 2005; Titsias, 2009) these introduce approximations that can distort the uncertainty prediction (Liu et al., 2020b).

More recently, several different approaches for uncertainty decomposition using deep learning frameworks have been proposed. Popular approaches rely on Bayesian approximations (Depeweg et al., 2017) or permanent dropouts (Gal, 2016). Furthermore, latent inputs can also be used to achieve a decomposition into aleatoric and epistemic uncertainty as presented by Depeweg et al. (2018). However, in particular for Bayesian NNs, these approaches become computationally challenging (Kwon et al., 2020). Furthermore, the prediction requires sampling the entire network before the statistics of the output can be computed. When applied to control problems that are real-time critical e.g., robotics with a sampling rate up to 1 kHz, these computational burdens prohibit an employment of these techniques. Ensemble-based approaches (Lakshminarayanan et al., 2017), that utilize an ensemble of individual models can also provide epistemic uncertainty estimates, but are also sample based and have significant memory requirements. A sampling-free estimation method 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 can become inaccurate depending on the non-linearity of the activation functions.

In contrast to the above approaches, we propose a deep learning framework that is able to estimate both epistemic and aleatoric uncertainty in real-time along with the model prediction without adding to the storage requirements of the model beyond that of an additional layer of neurons. Our method models aleatoric uncertainty by extending the neural network that models the dynamics with an additional head and incorporating the output into the training loss function similar to Kendall and Gal (2017). For estimating epistemic uncertainty, our method learns a classifier that can detect the absence of training data or in other words, detect whether a test sample is out-of-distribution (OOD). We evaluate the proposed method on low-dimensional synthetic and real-world benchmark datasets both qualitatively and quantitatively. To demonstrate the applicability of real-time uncertainty decomposition via our method, we employ the proposed epistemic uncertainty estimation method to learn a data-efficient controller for a simulated quadcopter navigating a space with unknown disturbances online while tuning controller gains in accordance with the aleatoric uncertainty estimates that our model outputs. The different treatment of the epistemic and aleatoric uncertainty measures is in contrast to previous works that take a general uncertainty measure for risk-averse control.

The remainder of this paper is structured as follows. In Sec. 2, we formalize our problem setting. Our approach for uncertainty decomposition for real-time control is de-

scribed in Sec. 4. We evaluate our approach in Sec. 5 before concluding the paper in Sec. 6.

2. PROBLEM FORMULATION

We consider the discrete-time dynamical system with control $\mathbf{u} \in \mathbb{U} \subseteq \mathbb{R}^{d_u}$ and state $\mathbf{x} \in \mathbb{X} \subseteq \mathbb{R}^{d_x}$

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

where $\mathbf{g}: \mathbb{X} \times \mathbb{U} \rightarrow \mathbb{X}$ is the known part of the system, while $\mathbf{y} \in \mathbb{Y} \subseteq \mathbb{R}^{d_y}$ represents a disturbance composed of the unmodeled part \mathbf{f} of the dynamics and a Gaussian distributed noise ϵ :

$$\mathbf{y}_k = \mathbf{f}(\mathbf{x}_k) + \epsilon, \quad \epsilon \in \mathcal{N}(0, \text{diag}(\boldsymbol{\sigma}(\mathbf{x}))), \quad (2)$$

where the diagonal noise covariance $\boldsymbol{\sigma} \in \mathbb{R}^{d_y}$ is unknown and cannot be measured independently of \mathbf{y} . While \mathbf{f} and $\boldsymbol{\sigma}$ are shown to be a function of \mathbf{x} only, it is straightforward to extend them to be a function of \mathbf{u} as well. However, we omit this for the sake of brevity.

Given that measurements can be taken to obtain the data set $\mathbb{D}_{\text{tr}} = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{N_{\text{tr}}}$ with inputs $\mathbb{X}_{\text{tr}} = \{\mathbf{x}_i\}_{i=1}^{N_{\text{tr}}}$ and outputs $\mathbb{Y}_{\text{tr}} = \{\mathbf{y}_i\}_{i=1}^{N_{\text{tr}}}$, our goal is to learn an uncertainty aware model of the stochastic process underlying \mathbf{y} .

Uncertainty in data-driven models arises from two distinct sources and is thus categorized into two types. The uncertainty in data, known as *aleatoric* uncertainty is inherent to the stochastic process underlying \mathbf{y} and is irreducible. However, with the knowledge that the disturbance is Gaussian distributed, one can model the parameters (mean and variance) of this distribution with a neural network with two outputs $[\hat{\mathbf{f}}, \hat{\boldsymbol{\sigma}}]$ (Kendall and Gal, 2017). The other type of uncertainty, termed *epistemic*, arises from the limited expressive power of the model or absence of sufficient data for training it (Gal, 2016).

In addition to modeling the true mapping \mathbf{f} with the neural network output $\hat{\mathbf{f}}: \mathbb{X} \rightarrow \mathbb{Y}$ and associated aleatoric uncertainty with $\hat{\boldsymbol{\sigma}}: \mathbb{X} \rightarrow \mathbb{R}_+^{d_y}$, our goal includes explicitly modeling the epistemic uncertainty of our learned model $\hat{\mathbf{f}}$ such that it can be estimated in real-time. More specifically, we assume η to be a mapping on \mathbb{X} that acts as an indicator of the model’s epistemic uncertainty at any input $\mathbf{x} \in \mathbb{X}$.

3. EPIOUT - DIRECT ESTIMATION OF EPISTEMIC UNCERTAINTY

Note that epistemic uncertainty at a $\mathbf{x} \in \mathbb{X}$ expresses the model $\hat{\mathbf{f}}$ ’s ignorance regarding the true mapping \mathbf{f} emanating from the absence of training data at (or close to) \mathbf{x} . We therefore posit that an OOD detector, which can indicate whether a sample is in training data distribution or not, can be used as a proxy for epistemic uncertainty.

We explicitly model this OOD detector via a separate module consisting of an output node and optionally distinct layers termed *EpiOut* added to the network with outputs $\hat{\mathbf{f}}$ and $\hat{\boldsymbol{\sigma}}$ (refer to Fig. 1 for an example schematic). Consequently, the dataset for training *EpiOut* must be constructed to reflect this absence of data. However, we do not use a held out dataset for learning this classifier.

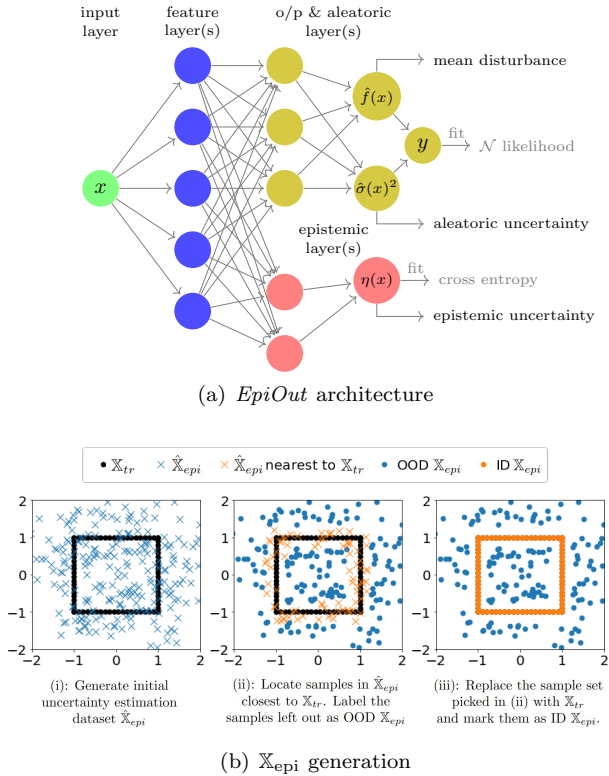


Fig. 1. Modeling and training *EpiOut*: (a) Schematic of an example Neural Network architecture with decomposed uncertainties (b) Process of generating samples \mathbb{X}_{epi} for learning the OOD classifier.

Instead, we generate OOD samples using a simple mechanism based on the hypothesis that OOD samples near the boundary of the training distribution are sufficient for learning an OOD detector that is able to generalize to the entire OOD space. Previous works by Lee et al. (2018); Marson et al. (2021) have backed up this hypothesis for classification problems. Similar to our work, Hafner et al. (2020) also proposes to generate OOD samples near the boundary of the training distribution for uncertainty estimation during regression. However, it simply adds probabilistic noise to the input to generate such samples which can result in overlap between the samples assigned OOD and training datapoints. In contrast, we follow a multi-step process that ensures the separation of the training distribution from samples assigned OOD (See Fig. 1 (b)).

In brief, (i) we first sample an initial set of points $\hat{\mathbb{X}}_{\text{epi}}$ close to the boundary of training samples, (ii) then locate the N_{tr} samples closest to the data and finally, (iii) replace this sample with \mathbb{X}_{tr} . We therefore generate a set \mathbb{X}_{epi} that contains both inputs in \mathbb{X}_{tr} labeled as low uncertainty data points as well as inputs in $\mathbb{X} \setminus \mathbb{X}_{\text{tr}}$, labeled as high uncertainty data points. We denote this epistemic dataset by $\mathbb{D}_{\text{epi}} = \{(\tilde{\mathbf{x}}_j, \tilde{y}_j)\}_{j=1}^{N_{\text{epi}}}$, $N_{\text{epi}} \in \mathbb{N}$, with inputs $\mathbb{X}_{\text{epi}} = \{\tilde{\mathbf{x}}_j\}_{j=1}^{N_{\text{epi}}}$ and outputs $\mathbb{Y}_{\text{epi}} = \{\tilde{y}_j\}_{j=1}^{N_{\text{epi}}}$.

Different variations for sampling the initial set $\hat{\mathbb{X}}_{\text{epi}}$ 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 $n \in \mathbb{N}$ points around each of the existing training points, that is a total of $N_{\text{epi}} = nN_{\text{tr}}$ from a normal distribution such that:

$$\hat{\mathbb{X}}_{\text{epi}} = \bigcup_{i=1}^{N_{\text{tr}}} \{\tilde{\mathbf{x}}_j \sim \mathcal{N}(\mathbf{x}_i, \text{diag}(\boldsymbol{\nu}))\}, j = 1, \dots, n\}, \quad (3)$$

where $\boldsymbol{\nu}$ is a parameter vector defining a diagonal covariance matrix. Hyper-parameters n and $\boldsymbol{\nu}$ must be judiciously chosen to ensure that we train *EpiOut* to be robust with respect to the unknown function \mathbf{x} we are attempting to model.

In the following, we lay out our proposals for the choice of these hyper-parameters and the intuitions behind them.

3.1 Choice of $\boldsymbol{\nu}$, n

$\boldsymbol{\nu}$ can be interpreted as an analogue to the lengthscale vector of a squared exponential kernel in a GP, that is, a measure for how far away from a training point the prediction is influenced: Larger $\boldsymbol{\nu}$ will lead to further spread of \mathbb{X}_{epi} and, therefore, low epistemic uncertainty in the neighborhood of the training data. A natural choice for $\boldsymbol{\nu}$ therefore can be the inverse of the gradient magnitude of the true mapping \mathbf{f} . If the gradient of the true mapping \mathbf{f} at a point \mathbf{x} in the input domain were large in magnitude, it would indicate that the value of $\mathbf{f}(\cdot)$ varies more quickly around the point \mathbf{x} as opposed to the case where this gradient were smaller. While the true mapping \mathbf{f} is unknown, a model $\hat{\mathbf{f}}$ trained on \mathbb{D}_{tr} to approximate \mathbf{f} is accessible. We therefore propose to choose $\boldsymbol{\nu}$ to be a function of the input point $\mathbf{x}_i \in \mathbb{X}_{\text{tr}}$ we sample around as well as the gradient of the trained model $\hat{\mathbf{f}}$. $\boldsymbol{\nu}$ can therefore be selected as:

$$\boldsymbol{\nu}(\mathbf{x}_i, \hat{\mathbf{f}}) = \mathbf{1}^{d_x} \oslash \left(\begin{bmatrix} \|\nabla_{\mathbf{x}} \hat{f}_1(\mathbf{x}_i)\|_{\infty} \\ \vdots \\ \|\nabla_{\mathbf{x}} \hat{f}_{d_x}(\mathbf{x}_i)\|_{\infty} \end{bmatrix} + c \cdot \mathbf{1}^{d_x} \right), \quad (4)$$

where $\mathbf{1}^{d_x}$ is a vector of size d_x with all elements equal to 1, \oslash is the element-wise (Hadamard) division operator and c is a positive constant whose choice bounds the maximum value of $\boldsymbol{\nu}$. It can generally be set to a small positive value $\ll 1$ in order to avoid division-by-zero errors, in the event the gradient at a point is 0. In an online learning setting, the value of c can be chosen to be large ≈ 1 when the number of training points is too low and $\nabla_{\mathbf{x}} \hat{\mathbf{f}}(\mathbf{x}_i)$, $\mathbf{x}_i \in \mathbb{X}_{\text{tr}}$ cannot be reliably estimated. This value can then be annealed to a small value $c \ll 1$ as the gradient estimate becomes more reliable. Finally, the uniform norm $\|\cdot\|_{\infty}$ selects the maximum element from a vector. Applying this to every column of the Jacobian matrix $\nabla_{\mathbf{x}} \hat{\mathbf{f}}$ yields a vector of d_x dimensions. Thus if the magnitude of the model's slope at a training point \mathbf{x} is high for one of the output dimensions with respect to a particular input dimension, we sample points for \mathbb{X}_{epi} closer to \mathbf{x} .

We propose setting n to a multiple of $2d_x + 1$. This corresponds to padding each training point in both directions of the input dimension with an OOD point on an average for the generation of the epistemic dataset \mathbb{D}_{epi} . The next

subsection discusses the final steps required for completing the creation of the epistemic dataset \mathbb{D}_{epi} .

3.2 Generation of \mathbb{Y}_{epi}

To define the set \mathbb{Y}_{epi} , we first compute the minimal distance (according to a distance metric $d: \mathbb{X} \times \mathbb{X} \rightarrow \mathbb{R}_{0,+}$) to the training data for each epi point.

$$d_j = \min_{\mathbf{x} \in \mathbb{X}_{\text{tr}}} d(\tilde{\mathbf{x}}_j, \mathbf{x}), \quad j = 1, \dots, N_{\text{epi}}. \quad (5)$$

Note that for an epi point $\tilde{\mathbf{x}}_j$, the closest training data point is not necessarily the one used to generate it. Let $d_{N_{\text{tr}}}$ be the N_{tr} -th smallest element of the set $\mathbf{d} = \{d_j\}_{j=1}^{N_{\text{epi}}}$.

We generate \mathbb{Y}_{epi} and derive \mathbb{X}_{epi} from $\hat{\mathbb{X}}_{\text{epi}}$ as follows

$$\tilde{\mathbf{y}}_j = \begin{cases} 1, & \text{if } d_j > d_{N_{\text{tr}}} \\ 0, & \tilde{\mathbf{x}}_j \leftarrow \arg \min_{\mathbf{x} \in \mathbb{X}_{\text{tr}}} d(\tilde{\mathbf{x}}_j, \mathbf{x}) \text{ if } d_j \leq d_{N_{\text{tr}}} \end{cases}. \quad (6)$$

The N_{tr} points in $\hat{\mathbb{X}}_{\text{epi}}$ with the least distance to a training point are therefore replaced by the corresponding point in \mathbb{X}_{tr} . This step attempts to reduce the overlap between the input space covered by $\mathbb{X}_{\text{epi}} \setminus \mathbb{X}_{\text{tr}}$ and the space covered by the training inputs \mathbb{X}_{tr} . Now the choice of $2d_x + 1$ epi points becomes clear as one of them will be turned into $\tilde{\mathbf{y}} = 0$ corresponding to “low epistemic uncertainty”, while $2d_x$ points further away from the training point $\tilde{\mathbf{y}} = 0$ correspond to “high epistemic uncertainty”. See Fig. 1(b) for an example of epistemic training data generation.

3.3 Overall architecture and training

Our uncertainty aware model for disturbance \mathbf{y} is implemented via a neural network with outputs $[\hat{\mathbf{f}}(\cdot) \hat{\boldsymbol{\sigma}}(\cdot) \eta(\cdot)]^T$, where $\hat{\mathbf{f}}(\cdot)$ corresponds to the underlying noiseless true mapping \mathbf{f} , $\hat{\boldsymbol{\sigma}}(\cdot)$ is the aleatoric uncertainty estimate, and, $\eta(\cdot)$ corresponds to the epistemic uncertainty estimate (see Fig. 1 for an illustration). The new output $\eta(\cdot)$ is terminated with a neuron using a sigmoidal activation function, such that $\eta: \mathbb{X} \rightarrow [0, 1]$. This is beneficial because it allows one to immediately judge whether the predicted uncertainty is *high* (≈ 1) or *low* (≈ 0) without any reference which would be required for unnormalized uncertainty quantifications.

We first train the feature layers as well as the output and the aleatoric uncertainty layers ($\hat{\mathbf{f}}, \hat{\boldsymbol{\sigma}}$) by minimizing a negative log likelihood loss with respect to training data \mathbb{D}_{tr} , similar to Kendall and Gal (2017). The gradient of the trained network can then be utilized to calculate $\boldsymbol{\nu}$ and sample the dataset \mathbb{D}_{epi} in accordance with the method laid out in the previous subsections.

The epistemic module (η) is trained independently of the loss function for the original network over the dataset \mathbb{D}_{epi} 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 simplicity, we use the Euclidean distance, however the method can be easily extended to other metrics. Since the number of epistemic data-points with uncertain labels (with value 1) are far greater than the ones with certain labels (with value 0), a class-weighted version of this loss is minimized.

3.4 Computational complexity

Generation of the pseudo data-set (3) is an operation with complexity $\mathcal{O}(N_{\text{epi}}) \hat{=} \mathcal{O}(N_{\text{tr}} d_x)$, whereas (5) is, for a trivial implementation, a $\mathcal{O}(N_{\text{tr}} N_{\text{epi}}) \hat{=} \mathcal{O}(d_x N_{\text{tr}}^2)$ operation. However, a kd-tree based implementation (Cormen et al., 2009) allows for an execution time complexity $\mathcal{O}(N_{\text{epi}} \log(N_{\text{epi}})) \hat{=} \mathcal{O}(d_x N_{\text{tr}} \log(N_{\text{tr}} d_x))$. Finding the N_{tr} smallest distances from all N_{epi} points in (6) can be obtained in $\mathcal{O}(N_{\text{tr}} + (N_{\text{epi}} - N_{\text{tr}}) \log(N_{\text{tr}})) \hat{=} \mathcal{O}(N_{\text{tr}} + N_{\text{tr}}(d_x - 1) \log(N_{\text{tr}}))$ time. The training of a neural network with a fixed number of weights requires $\mathcal{O}(N_{\text{epi}}) \hat{=} \mathcal{O}(N_{\text{tr}} d_x)$. Hence, the overall complexity results in $\mathcal{O}(d_x N_{\text{tr}} \log(d_x N_{\text{tr}}))$, and the overall storage complexity is given by $\mathcal{O}(N_{\text{epi}} d_x) \hat{=} \mathcal{O}(N_{\text{tr}} d_x^2)$ for storing the set \mathbb{X}_{epi} . When considering streaming data (as for online learning control), the set \mathbb{D}_{epi} can be constructed iteratively, reducing the complexity to $\mathcal{O}(\log(N_{\text{tr}}))$

4. REAL-TIME CONTROL WITH DECOMPOSED UNCERTAINTIES

We can now deploy the proposed approach for uncertainty decomposition in a real-time control loop. We utilize the *EpiOut* module and its output η for adding new data points for combating epistemic uncertainty in a data-efficient manner, while exploiting the aleatoric uncertainty estimates $\hat{\boldsymbol{\sigma}}$ for adjusting the controller gain to measurement noise. The latter robustifies the closed-loop control mechanism against process noise and can even guarantee stability (Beckers et al., 2019), while keeping the energy consumption low. The former combats epistemic uncertainty via increased data collection.

Considering the dynamical system given in (1), our task is to choose a control input \mathbf{u} at each step, such that the system follows a given reference \mathbf{x}^{des} . Furthermore, the controller can take a new measurement \mathbf{y} and add $\{\mathbf{x}, \mathbf{y}\}$ to the training dataset to improve its model over time but each such measurement is considered costly and therefore new training points should only be collected when necessary. This is typical in distributed systems, where multiple sensors share the same scarce communication channel, or in autonomous systems with limited data storage capacity.

4.1 Epistemic uncertainty utilization for data-efficient model learning

The need for high data efficiency requires models that can judge their own fidelity in real-time to identify valuable measurements. We utilize *EpiOut* by adding new measurements $(\mathbf{x}_i, \mathbf{y}_i)$ to the training dataset based on the epistemic uncertainty $\eta(\cdot) \in [0, 1]$ of the data-point. More concretely, we use the epistemic uncertainty to parametrize a Bernoulli distribution which is used to determine whether to add the measurement to the training dataset or not,

$$\mathbb{D}_{\text{tr}} \leftarrow \mathbb{D}_{\text{tr}} \cup \begin{cases} (\mathbf{x}_i, \mathbf{y}_i) & \text{if } \xi = 1 \\ \emptyset & \text{if } \xi = 0 \end{cases}, \quad (7)$$

where $\xi \sim \mathcal{B}(\alpha)$, $\alpha = \eta(\mathbf{x}_i)$.

This ensures the high accuracy of the disturbance model $\hat{\mathbf{f}}$ while at the same time bolstering data-efficiency in the online data collection process, as training data is added

only when necessary, thus reducing the number of costly measurements.

4.2 Aleatoric uncertainty based gain adjustment

The system (1) is inherently random due to the stochastic nature of the disturbance \mathbf{y} . Therefore, we combine feedback and a feedforward control law

$$\mathbf{u} = \mathbf{K} (\mathbf{x} - \mathbf{x}^{\text{des}}) + \mathbf{u}_{\text{ff}}, \quad (8)$$

where \mathbf{u}_{ff} is a feedforward control term determined based on the known model $\mathbf{g}(\cdot, \cdot)$ and the learned disturbance model $\hat{\mathbf{f}}(\cdot)$. The choice of the feedback gain \mathbf{K} , which compensates for imprecision in the model and the stochasticity of the disturbance, is difficult because high control gains lead to high tracking performance, but also consume more energy and amplify measurement noise, which can lead to increased tracking errors. It is therefore, generally advisable to let the feedforward term \mathbf{u}_{ff} take over most of the control effort and keep the feedback term small when the model is reliable (see Beckers et al. (2019)). We thus increase the gains only if the aleatoric uncertainty inferred by our model as $\hat{\sigma}(\cdot)$ is high. The feedback gains can therefore be adjusted as:

$$\mathbf{K} = \bar{\mathbf{K}} (1 + \beta \|\hat{\sigma}(\mathbf{x})\|_{\infty}), \quad (9)$$

where $\beta \in \mathbb{R}_{0,+}$ is the sensitivity and $\bar{\mathbf{K}} \in \mathbb{R}_{+}^{d_u \times d_x}$ defines the minimal control gain.

Therefore, we tune the feedback gains only based on the aleatoric uncertainty, while combating the epistemic uncertainty with an increased data collection rate (7).

5. EVALUATION

We evaluate our proposed approach for estimating epistemic uncertainty both qualitatively and quantitatively for a variety of synthetic and real datasets and compare them to existing approaches. We then demonstrate the utility of our decomposable uncertainty estimation approach for data-efficient online learning and control on a simulated quadcopter system affected by unknown disturbances.

5.1 Qualitative and Quantitative Evaluation for Regression

Our approach is evaluated on two simulated and one real-world dataset for illustrating the advantage of our approach when estimating uncertainty due to absence of data. Both are 1-dimensional regression tasks. The first, *1D Split*, has the nominal function $f(x) = \sin(x\pi)$, with training points located only in 2 bands $\mathbb{X}_{\text{tr}} = \{x_i \sim \mathcal{U}(-2, -1)\}_{i=1}^{100} \cup \{x_i \sim \mathcal{U}(1, 2)\}_{i=101}^{200}$ and $N_{\text{te}} = 961$ test points are placed on a grid $[-4, 4]$. The second, *2D Gaussian* ($d_x = 2$, $d_p = 1$) has the nominal function $f(\mathbf{x}) = \frac{\sin(5x_1)}{5x_1} + x_2^2$. Two clusters of 500 points each are respectively sampled from Gaussian distributions placed around points $(0, -1)$ and $(0, 1)$. 961 test points are uniformly placed on a grid $[-2, 2]^2$. We additionally evaluate our approach on a real dataset *Sarcos* (Vijayakumar and Schaal, 2000) which records the inverse dynamics of a seven degrees-of-freedom SARCOS anthropomorphic robot arm $d_x = 21$, $d_p = 1$. 10000 training samples and 2000 test samples were randomly extracted from a total of 5×10^4 samples.

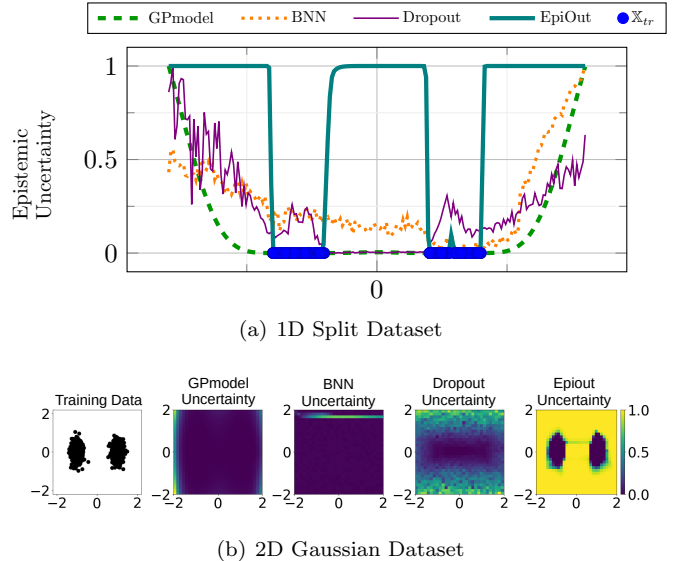


Fig. 2. The epistemic uncertainty given by different models for datasets (a) 1D Split and (b) 2D Gaussian. Note that unlike EpiOut, other methods are unable to predict a higher uncertainty between the training data clusters.

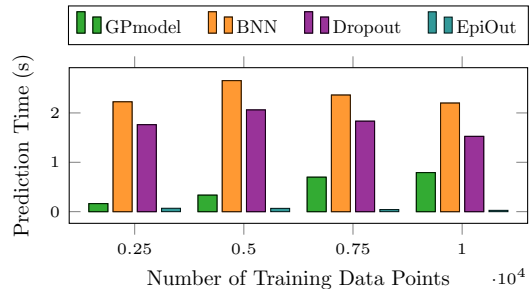


Fig. 3. The prediction time for various models with increasing training data (Sarcos). As is evident, *EpiOut* outperforms all other models in this regard.

Table 1. Mean Standardized Log loss for the considered models across datasets.

	1D Split	2D Gaussian	Sarcos
GPmodel	-3.69	873.21	0.1534
BNN	0.73	-75.6	-1.37
Dropout	2603	2.34	-1.65
EpiOut	-0.08	-76.7	-1.74

We compare our approach to several state-of-the-art methods. These include a Gaussian Process (*GPmodel*) with a squared exponential automatic relevance determination kernel, a Bayesian Neural Network (*BNN*) with 2 fully connected hidden layers each with 50 hidden units and normal distributions over their weights, a neural network with 2 fully connected permanent layers each with 50 hidden units with dropout probability $\rho = 0.05$ (*Dropout*). Our proposed model *EpiOut* has 2 fully connected layers (50 neurons), $n = 3$ and ν as given by (4), where the value c is set to 10^{-5} .

5.2 Results & Discussion

Evaluation of the proposed method using Mean Standardized Log Likelihood (MSLL) (Rasmussen and Williams, 2006) is reported in Table 1. Since our approach predicts values for uncertainty constrained between $[0, 1]$, for fair comparison we calculated MSLL for *EpiOut*, by scaling the uncertainty estimate by a scalar value that is optimized to maximize the log-likelihood of the training data. We note that *EpiOut* demonstrates competitive performance when evaluated using MSLL for most of the datasets.

Additionally, an illustration of epistemic uncertainty for datasets *1D Split* and *2D Gaussian* for all models is shown in Fig. 2. Note that, in contrast to other approaches, *EpiOut* is more successful in identifying out of distribution data. We further point out additional benefits of our approach: (i) The *EpiOut* model predicts the uncertainty measure in a sample-free manner and as a result, in real time. This is typically an order of magnitude faster than *Dropout* and *BNN* (see Fig. 3). This is crucial for data-efficient online learning, where the epistemic uncertainty estimate is used to evaluate the usefulness of an incoming data point and is thus calculated more frequently than the online training is executed. (ii) 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 provide a return value $[0, \infty]$, which can be difficult to interpret without a maximum value as reference.

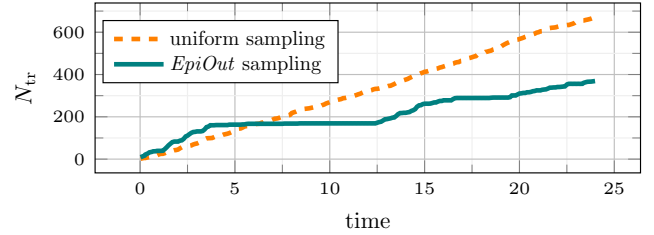
5.3 Event-triggered Online Learning Control Using Uncertainty Decomposition

As an application of our proposed approach, we consider the task of controlling a quadcopter with known dynamics $\mathbf{g}(\cdot, \cdot)$ (1), which explores a given terrain with unknown thermals \mathbf{y}^1 . Thus \mathbf{y} is an indicator of the thermal disturbance on the quadcopter acceleration in vertical direction. We assume that it follows the underlying distribution (2).

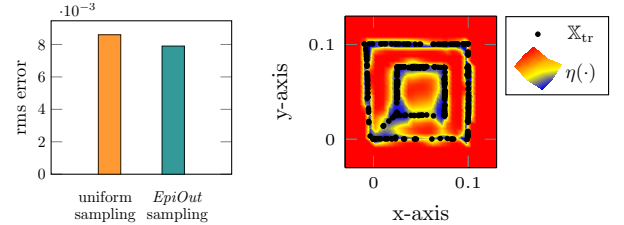
The task here is to execute the desired trajectory \mathbf{x}^{des} which we specify as three rounds at constant height on a square in the horizontal plane with edge length 0.1 followed by three rounds with edge length 0.05.

Following the method laid out in section 4 for implementing a data-efficient control strategy, the tracking performance of the proposed quadcopter controller, the data collections rate and the epistemic uncertainty model are illustrated in Fig. 4. The *EpiOut* prediction of the disturbance model allows smart selection of the measurements taken for training. While a uniformly random selection constantly takes data points, as shown in Fig. 4(a), our proposed approach takes more measurements when entering new areas (0s to 5s and 12s to 17s) and less when tracking the path already traversed. This is in accordance with the epistemic uncertainty estimate η which is low near training data and high in unobserved regions as is observable in Fig. 4(c). As illustrated in Fig. 4(b), this results in overall competitive tracking performance of a

¹ The data of the thermals is taken from publicly available paragliding data <https://thermal.kk7.ch>. Thermals or thermal columns are created by the uneven heating of the earth's surface and cause a vertical air flow due to convection (Ákos et al., 2010).



(a) Number of training datapoints acquired when sampling using *EpiOut* vs via a uniform distribution



(b) Root Mean Square Error during tracking (c) Epistemic uncertainty estimate ($\eta \in [0, 1]$) over the horizontal plane

Fig. 4. Deploying the control strategy from section 4 with *EpiOut* results in data-efficiency (a) while demonstrating competitive performance to the method which utilizes all available data for training (b). Epistemic uncertainty over the horizontal plane, indicated by absence of training data and estimated at the end of the task is shown in (c).

root mean squared error in z-direction of 0.0079 vs 0.0086 while using and storing less data.

6. CONCLUSION

This paper presents a novel deep learning structure for decomposing epistemic and aleatoric uncertainty and proposes a control framework that distinguishes between the use of these uncertainty measures. As the predictions and uncertainty estimates are obtained in a sample-free manner, the method can be utilized for real-time critical online learning and shows competitive performance against existing methods. The proposed online learning control algorithm is not only inherently data-efficient by adding only those data-points to the data set that can reduce the model's epistemic uncertainty, but is additionally able to provide a means to adapt the control gains for better tracking performance by taking the aleatoric uncertainty estimates into consideration.

For future work, we will consider alternative functions for sorting the epi points to encode prior knowledge, such as periodicity (similar to a kernel of GP) and investigate the effect of a continuous-valued epistemic uncertainty label \mathbb{Y}_{epi} . Additionally, we will explore the application of our uncertainty estimation approach for different contexts, for example in scenarios where instead of unknown system dynamics, other aspects such as part of the cost or system constraints are unknown.

REFERENCES

- Ákos, Z., Nagy, M., Leven, S., and Vicsek, T. (2010). Thermal soaring flight of birds and unmanned aerial vehicles. *Bioinspiration & Biomimetics*, 5(4), 045003.

- Beckers, T., Kulić, D., and Hirche, S. (2019). Stable Gaussian process based tracking control of Euler-Lagrange systems. *Automatica*, 23(103), 390–397.
- Capone, A., Lederer, A., Umlauf, J., and Hirche, S. (2020). Data selection for multi-task learning under dynamic constraints. *IEEE Control Systems Letters*, 5(3), 959–964.
- Cormen, T.H., Leiserson, C.E., Rivest, R.L., and Stein, C. (2009). *Introduction to algorithms*. MIT press.
- Depeweg, S., Hernández-Lobato, J.M., Doshi-Velez, F., and Udluft, S. (2017). Learning and policy search in stochastic dynamical systems with Bayesian neural networks. In *International Conference on Learning Representations*.
- Depeweg, S., Hernandez-Lobato, J.M., Doshi-Velez, F., and Udluft, S. (2018). Decomposition of uncertainty in Bayesian deep learning for efficient and risk-sensitive learning. In *International Conference on Machine Learning*.
- Gal, Y. (2016). *Uncertainty in Deep Learning*. Ph.D. thesis, University of Cambridge.
- Grigorescu, S., Trasnea, B., Cocias, T., and Macesanu, G. (2020). A survey of deep learning techniques for autonomous driving. *Journal of Field Robotics*, 37(3), 362–386.
- Hafner, D., Tran, D., Lillicrap, T., Irpan, A., and Davidson, J. (2020). Noise contrastive priors for functional uncertainty. In *Uncertainty in Artificial Intelligence*, 905–914. PMLR.
- Kendall, A. and Gal, Y. (2017). What uncertainties do we need in Bayesian deep learning for computer vision? In *Advances in Neural Information Processing Systems (NeurIPS)*, 5574–5584.
- Kwon, Y., Won, J.H., Kim, B.J., and Paik, M.C. (2020). Uncertainty quantification using Bayesian neural networks in classification: Application to biomedical image segmentation.
- Lakshminarayanan, B., Pritzel, A., and Blundell, C. (2017). Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in Neural Information Processing Systems (NeurIPS)*, 6402–6413.
- Lee, K., Lee, H., Lee, K., and Shin, J. (2018). Training confidence-calibrated classifiers for detecting out-of-distribution samples. In *International Conference on Learning Representations*.
- Liu, A., Shi, G., Chung, S.J., Anandkumar, A., and Yue, Y. (2020a). Robust regression for safe exploration in control. In *Conference on Learning for Dynamics and Control (L4DC)*.
- Liu, H., Ong, Y.S., Shen, X., and Cai, J. (2020b). When Gaussian process meets big data: A review of scalable gps. *IEEE Transactions on Neural Networks and Learning Systems*, 31(11), 4405–4423.
- Marson, L., Li, V., and Maki, A. (2021). Boundary optimised samples training for detecting out-of-distribution images. In *2020 25th International Conference on Pattern Recognition (ICPR)*, 10486–10492. IEEE.
- Postels, J., Ferroni, F., Coskun, H., Navab, N., and Tombari, F. (2019). Sampling-free epistemic uncertainty estimation using approximated variance propagation. In *Proceedings of the IEEE International Conference on Computer Vision*, 2931–2940.
- Quinero-Candela, J. and Rasmussen, C.E. (2005). A unifying view of sparse approximate Gaussian process regression. *The Journal of Machine Learning Research*, 6, 1939–1959.
- Rasmussen, C.E. and Williams, C.K.I. (2006). *Gaussian Processes for Machine Learning*. MIT Press, Cambridge, MA, USA.
- Snelson, E. and Ghahramani, Z. (2005). Sparse Gaussian processes using pseudo-inputs. *Advances in Neural Information Processing Systems*, 18.
- Titsias, M. (2009). Variational learning of inducing variables in sparse Gaussian processes. In *Artificial intelligence and statistics*, 567–574. PMLR.
- Umlauf, J. and Hirche, S. (2020). Feedback linearization based on Gaussian processes with event-triggered online learning. *IEEE Transactions on Automatic Control (TAC)*, 1–16.
- Umlauf, J., Pöhler, L., and Hirche, S. (2018). An uncertainty-based control Lyapunov approach for control-affine systems modeled by Gaussian process. *IEEE Control Systems Letters*, 2(3), 483–488.
- Vijayakumar, S.S. and Schaal, S. (2000). Locally weighted projection regression: An O(n) algorithm for incremental real time learning in high dimensional space. In *International Conference on Machine Learning (ICML)*.