Uncertainty in Equation Learning

Matthias Werner*
University of Tübingen
Tübingen, Germany

Andrej Junginger
ETAS GmbH, Bosch Group
Stuttgart, Germany

Philipp Hennig
University of Tübingen
Tübingen, Germany

Georg Martius
Max Planck Institute for Intelligent Systems
Tübingen, Germany

ABSTRACT

Equation learning is a deep learning framework for explainable machine learning in regression settings, with applications in engineering and the natural sciences. Equation learners typically do not capture uncertainty about the model or its predictions, although uncertainty is often highly structured and particularly relevant for these kinds of applications. We show how simple, yet effective forms of Bayesian deep learning can be used to build structure and explainable uncertainty over a set of found equations. Specifically, we use a mixture of Laplace approximations, where each mixture component captures a different equation structure, and the local Laplace approximations capture parametric uncertainty within one family of equations. We present results on both synthetic and real world examples.

KEYWORDS

uncertainty quantification, symbolic regression, equation learning

ACM Reference Format:
Matthias Werner, Andrej Junginger, Philipp Hennig, and Georg Martius. 2022. Uncertainty in Equation Learning. In *Genetic and Evolutionary Computation Conference Companion (GECCO '22 Companion), July 9–13, 2022, Boston, MA, USA.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3520304.3533964

1 INTRODUCTION

Equations are key elements in natural sciences to describe phenomena and their underlying principles. Similarly, they are important in the engineering domain, e.g., in model-predictive control [8]. In industrial applications, e.g., for embedded controller, models have to be minimal in computational power and memory demand due to embedded hardware and latency constraints. Model predictions given by equations can meet those requirements. Equation learning is a topic of growing interest in machine learning [11, 26, 29, 31]. In active learning, safe reinforcement learning and extrapolation tasks as well as safety critical systems like health care or automated driving, it is crucial to know about the uncertainty of the model. But, equation learners do not capture uncertainty about the model output. An equation learner discovers a set of plausible equations with different structure and varying complexity. Those equations correspond to different minima in the loss landscape. Their differences in structure can lead to a rich variety of predictions. Motivated by Occam's Razor [22] to prefer simple solutions, conventional equation learners select one equation, ignoring all others. However, if the dataset is ambiguous then there exist several similarly plausible candidate equations. Selecting one equation out of the others by chance means a loss of information about the data. This leads to overconfidence in favor of the selected equation. State-of-the-art in uncertainty quantification in deep learning are ensemble methods [6, 10, 16, 23, 28], especially to capture *global uncertainty* about the models predictions. So the question arises whether we can use the found equations by an equation learner to construct an ensemble of equations for uncertainty quantification in equation learning. Due to their differences in structure, the predictions of the equations strongly differ in regions that are not sufficiently covered by measurements. This is of special interest in extrapolation tasks, where no train data is available. Indeed, incorporating equations of different complexity and structure leads to sophisticated, structured uncertainty estimates. We apply a Laplace approximation to each equation in order to capture *local uncertainty* about the parameters within each equation structure. Great success has recently been shown for uncertainty quantification in deep learning [3, 7, 9, 13, 14, 30].

We show how simple, yet effective forms of Bayesian deep learning can be used to build such structured (explainable) uncertainty over a set of found equations. Specifically, we use a mixture of Laplace approximations, where each mixture component captures a different equation structure, and the local Laplace approximations capture parametric uncertainty within one family of equations.

The paper is structured as follows: First, we outline background information about Gaussian regression and equation learning in section 2. In section 3 we introduce our method to capture uncertainty in equation learning. In section 4, we present applications of our method to two artificial, ambiguous datasets as well as two real-world datasets. In section 5 we discuss relations to other work and conclude in section 6.

*corresponding author: m_werner@posteo.de

GECCO '22 Companion, July 9–13, 2022, Boston, MA, USA

ACM ISBN 978-1-4503-9268-6/22/07…$15.00
https://doi.org/10.1145/3520304.3533964

Table 1: Hand-picked equations of different complexity for toy example E1 and E2 with root mean squared error (rmse) and mixture coefficients π_k on the train dataset. Their local Laplace approximations are shown in the middle panel of figure 1.

dataset	π_k	rmse	complexity	equation
E1	0.095	0.0098	2	$y_0 = 0.79 - 0.34\,(1.05x_1 + 0.03)^2$
	0.063	0.0088	2	$y_0 = 0.56 - 0.24\cos(2.02x_1 + 3.16)$
	0.041	0.0043	4	$y_0 = -0.45\,(-0.98x_1 - 0.04)^2 + 0.02\cos(3.92x_1 - 2.2) + 0.81$
	0.012	0.00021	8	$y_0 = -0.15\,(0.85x_1 - 0.04)^2 - 0.02\cos(2.54x_1 + 3.08)$ $-0.04\cos(2.62x_1 - 3.2) - 0.07\cos\left(7.78\,(0.85x_1 - 0.04)^2 - 3.64\right) + 0.67$
E2	0.057	0.0088	3	$y_0 = 0.44 - 0.7\left(0.3 - 1.24\,(1.88x_1 + 0.06)^2\right)^2$
	0.051	0.0084	3	$y_0 = 0.45 - 0.45\,(0.5 - 0.98\cos(4.03x_1 + 0.13))^2$
	0.049	0.038	2	$y_0 = 0.43 - 0.52\,(-1.15x_1 - 0.05)^2$
	0.014	7.8e-06	4	$y_0 = 0.82 - 0.6\,(-0.01x_1 + 1.16\cos(7.54x_1 + 0.26) + 0.2)^2$

2 BACKGROUND

2.1 Gaussian regression

In this work, we consider Gaussian regression with a parameterized analytical equation $f_{\boldsymbol{\theta}}$, to map a d-dimensional input $\boldsymbol{x}$ to a d'-dimensional output $\boldsymbol{y}$ with posterior distribution

$$p(\boldsymbol{\theta} \mid \mathcal{D}) \propto p(\mathcal{D} \mid f_{\boldsymbol{\theta}}) p(\boldsymbol{\theta}) = \prod_{i=1}^{N} \mathcal{N}(\boldsymbol{y}_i \mid f_{\boldsymbol{\theta}}(\boldsymbol{x}_i), \sigma^2) \mathcal{N}(\boldsymbol{\theta} \mid 0, \lambda^{-1} I) \quad (1)$$

with a Gaussian zero-mean isotropic prior $p(\boldsymbol{\theta})$ on the parameters $\boldsymbol{\theta}$ of the equation, which corresponds to L_2 regularization on the parameters with a scalar precision hyperparameter λ. The dataset $(\boldsymbol{x}_i, \boldsymbol{y}_i) \in \mathcal{D}$ is assumed to be sampled iid., and it contains N data points each of which has Gaussian noise with variance σ^2. The empirical risk for an equation with M parameters is given by the negative log-posterior

$$L(\mathcal{D}, f_{\tilde{\boldsymbol{\theta}}}) = \frac{\lambda}{2} \sum_{i=1}^{M} \tilde{\theta}_i^2 + \frac{M}{2} \log(2\pi\lambda^{-1}) + \sum_{i=1}^{N} \frac{(\boldsymbol{y}_i - f_{\tilde{\boldsymbol{\theta}}}(\boldsymbol{x}_i))^2}{2\sigma^2} + \frac{N}{2} \log(2\pi\sigma^2). \quad (2)$$

We assume an equation learner infers the structure of $f_{\boldsymbol{\theta}}$ as well as its parameters from data.

2.2 Equation learning

Learning simple and accurate equations for a given dataset is known as *equation learning* or *symbolic regression*. The found equations are explainable and describe the relations between the quantities of interest. They are compositions of mathematical building blocks like analytic functions $\{\sin, \text{sqrt}, \log, \dots\}$, mathematical operators like $\{+, -, *, /, \circ, \dots\}$ and constants. Our framework to capture uncertainty in equation learning can be applied to any set of equations found by an equation learning algorithm.

3 THEORY

We capture the parametric uncertainty within one equation with a local Laplace approximation and refer to it as *local uncertainty*. Yet, an equation learner finds several plausible equations of varying structure and complexity. They correspond to different modes of the likelihood. Due to their differences in structure their prediction strongly deviates from each other in regions that are not sufficiently covered by measurement data. This applies especially to extrapolation tasks to unseen domains. We describe this variety in predictions as *global uncertainty* that we capture with a mixture of local Laplace approximations (MoLA, Eschenhagen et al. [6]).

3.1 Linearized Laplace approximation for equations

The Laplace approximation can be applied *post-hoc* to pre-trained models, e.g., to learned equations f. It approximates a distribution at its mode by matching a multivariate Gaussian to that mode with a covariance given by the inverse of the curvature at the mode. Here, it approximates the posterior $p(\boldsymbol{\theta} \mid \mathcal{D})$ at its mode $\tilde{\boldsymbol{\theta}}$ by a multivariate Gaussian [22]

$$p(\boldsymbol{\theta} \mid \mathcal{D}) \approx \mathcal{N}(\boldsymbol{\theta} \mid \tilde{\boldsymbol{\theta}}, \boldsymbol{A}^{-1}) \quad (3)$$

with the inverse curvature of the negative log-posterior

$$\boldsymbol{A} = -\nabla\nabla^\intercal \ln(p(\mathcal{D} \mid \boldsymbol{\theta}))\,|_{\tilde{\boldsymbol{\theta}}} - \nabla\nabla^\intercal \ln(p(\boldsymbol{\theta})\,|)_{\tilde{\boldsymbol{\theta}}}\,. \quad (4)$$

The first term is the Hessian of the negative log-likelihood $\boldsymbol{H}$. The second term corresponds to the prior. The predictive distribution of $f_{\boldsymbol{\theta}}^*$ at any test point $\boldsymbol{x}^*$ is given by integration over the weights

$$p(\boldsymbol{y}^* | \boldsymbol{x}^*, \mathcal{D}) = \int p(\boldsymbol{y}^* \mid f_{\boldsymbol{\theta}}(\boldsymbol{x}^*)) p(\boldsymbol{\theta} \mid \mathcal{D}) \mathrm{d}\boldsymbol{\theta}\,. \quad (5)$$

The integration is typically intractable. It can be approximated either by Monte-Carlo integration from parameter distribution $p(\boldsymbol{\theta} \mid \mathcal{D})$ or with a linearization at its mode $\tilde{\boldsymbol{\theta}}$

$$f_{\boldsymbol{\theta}}(\boldsymbol{x}^*) \approx f_{\tilde{\boldsymbol{\theta}}}(\boldsymbol{x}^*) + \boldsymbol{J}^{*\intercal}(\boldsymbol{\theta} - \tilde{\boldsymbol{\theta}}) \quad (6)$$

with Jacobian $\boldsymbol{J}^* = \nabla_{\boldsymbol{\theta}} f_{\boldsymbol{\theta}}(\boldsymbol{x}^*)|_{\tilde{\boldsymbol{\theta}}}$. This leads to a tractable predictive Gaussian distribution whose mean is given by the found equation $f_{\tilde{\boldsymbol{\theta}}}$ with parameters set to its mode $\tilde{\boldsymbol{\theta}}$

$$p(\boldsymbol{y}^* \mid \boldsymbol{x}^*, \mathcal{D}) \approx \mathcal{N}(\boldsymbol{y}^* \mid f_{\tilde{\boldsymbol{\theta}}}(\boldsymbol{x}^*), \Sigma^*), \quad \Sigma^* = \boldsymbol{J}^{*\intercal} \boldsymbol{A}^{-1} \boldsymbol{J}^* + \sigma^2 \boldsymbol{I}\,. \quad (7)$$

The first term in the covariance Σ depends on $\boldsymbol{x}$ and describes the uncertainty of a certain equation structure due to uncertainty of the parameters $\boldsymbol{\theta}$. The second term is given by measurement uncertainty.

3.2 Mixture of Laplace approximations

We capture the global uncertainty of K equations with a mixture of Laplace approximations (MoLA)

$$p(\boldsymbol{y}^* \mid \boldsymbol{x}^*, \mathcal{D}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\boldsymbol{y}^* | f_{\hat{\boldsymbol{\theta}}}^k(\boldsymbol{x}^*)), \Sigma_k^*) \tag{8}$$

The mixture coefficients are chosen such that they reflect how plausible each equation is w.r.t. to the data. This should be related to accuracy and complexity of the equation. From a Bayesian view, the natural choice is the marginal likelihood since it inherits a regularization for complexity and accuracy

$$p(\mathcal{D} \mid f^k) = \int p(\mathcal{D} \mid f_{\boldsymbol{\theta}}^k) p(\boldsymbol{\theta}) \, \mathrm{d}\boldsymbol{\theta} \approx \mathrm{e}^{-L(\mathcal{D}, f_{\hat{\boldsymbol{\theta}}}^k)} \sqrt{\frac{(2\pi)^{M_k}}{\det \boldsymbol{A}}} \tag{9}$$

with the empirical risk L (see equation 2) given by the negative log-posterior and the determinant of the curvature matrix $\boldsymbol{A}$. Small Eigenvalues of $\boldsymbol{A}$ increase the marginal likelihood. They indicate directions in weight space that are not identified by data and thus represent flexibility of the equation in parameter space. The marginal likelihood can be decomposed in two contributions, the empirical loss favors accurate, yet simple equations and the determinant of the curvature favors flexible equations in parameter space. The marginal likelihood is relevant in Bayesian model selection. It is a measure to compare different models to each other. Typically, hyperparameters are optimized w.r.t. to the marginal likelihood. Therefore, we choose the mixture coefficients π_k to be proportional to the corresponding normalized marginal likelihood

$$\pi_k = \left(p(\mathcal{D} \mid f^k)\right)^{1/N} / Z, \qquad Z = \sum_{k=1}^{K} \left(p(\mathcal{D} \mid f^k)\right)^{1/N} \tag{10}$$

In contrast to classic equation learning, which chooses one specific equation, our multi-modal mixture model captures several plausible equations and their local parametric uncertainty. For fast evaluation we approximate the predictive distribution of the mixture of Laplace approximations with its first two moments similar to Lakshminarayanan et al. [16]

$$\boldsymbol{m}^* = \sum_{k=0}^{K} \pi_k f_{\hat{\boldsymbol{\theta}}}^k(\boldsymbol{x}^*) \tag{11}$$

$$\operatorname{diag} \Sigma^* = \sum_{k=0}^{K} \pi_k (\operatorname{diag} \Sigma_k^* + \boldsymbol{y}_k^{*2}) - \boldsymbol{m}^{*2} \,. \tag{12}$$

This leads to the following Gaussian distribution

$$p(\boldsymbol{y}^* \mid \boldsymbol{x}^*, \mathcal{D}) = \mathcal{N}(\boldsymbol{y}^* \mid \boldsymbol{m}^*, \operatorname{diag} \Sigma^*). \tag{13}$$

We want to stress that the computational overhead is still small compared to neural networks, since just one forward pass and one backward pass of each equation is required to calculate the approximation. A visualization is shown in right panel of figure 4.

Remarks: The conditional mean of multi modal distributions can give a poor representation of the data. Especially, in extrapolation regions where the predictions of the underlying equation components strongly deviate. Depending on the application the conditional mode may be more meaningful. It would require numerical iteration since it has no analytic solution.

Our mixture of Laplace approximations strongly depends on the set of found equations by an equation learner. It inherits a bias towards structures of equations that occur more frequently. This might be wanted in the sense that the equation learner introduces a bias upon the structure of found equations.

4 EXPERIMENTS

In this section, we investigate the predictive distribution of the MoLA with its local Laplace approximations. We highlight the importance of considering several plausible equations instead of one equation on the basis of two ambiguous toy datasets in section 4.1. In section 4.2 we study two real world time series, which have been investigated in the course of structured uncertainty estimates by the *automatic statistician* [20].

Training: We normalize input and output of the real world datasets for training. We use the equation learner iEQL of Werner et al. [36] to retrieve plausible functions, which describe the datasets. More technical details are given in appendix A.1.

We refer to complexity in the framework of the iEQL. It measures the complexity of an equation by counting all active weights that are necessary to represent the equation.

Hyperparameters: We follow common practice to optimize the precision λ by maximizing the marginal likelihood. Therefore, we apply a grid search with $\lambda = 10^k$ where k is in the range from -2 to 2 with 100 equally spaced steps. The scale of the likelihood σ^2 is a model of measurement error. In technical applications, this parameter is usually known as part of the calibration of the measurement process and should then not be estimated. In situations where it is not known it can be estimated *post-hoc* empirically as $\sigma^2 = \sum_i^N (f_{\boldsymbol{\theta}}(\boldsymbol{x}_i) - \boldsymbol{y}_i)^2 / N$, with the usual risk of model-overfitting.

Remarks: During our experiments we discovered that for deep neural networks and also for the considered equations the Hessian is not guaranteed to be positive-semi-definite (psd.) after convergence of the optimizer. This might be due to the use of the Adam[12] optimizer, which can converge to a saddle point with some directions with negative curvatures. Therefore, we approximate the Hessian with the generalized Gauss-Newton (ggn) matrix with backpack for pytorch [2], which is positive-semi-definite by construction. The use of a ggn approximated Hessian is motivated by the findings of Immer et al. [9], since we are using a linearized Laplace approximation.

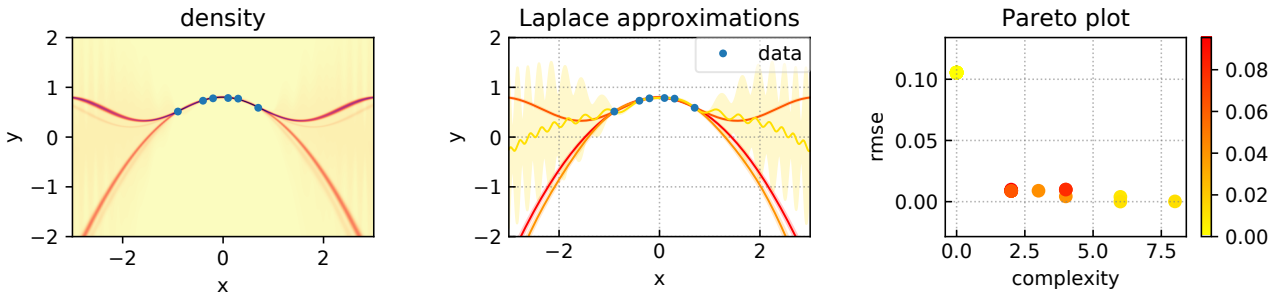
4.1 Illustrative toy examples

In order to highlight the importance of considering several plausible equations instead of one single equation as in conventional equation learning, we construct two ambiguous datasets in which , without further knowledge, polynomial and periodic equations are similarly plausible. Selecting one equation out of the others by chance means a loss of information about the data. The ground truth functions are chosen such, that they resemble a second order polynomial despite having a cos structure with different measurement noise

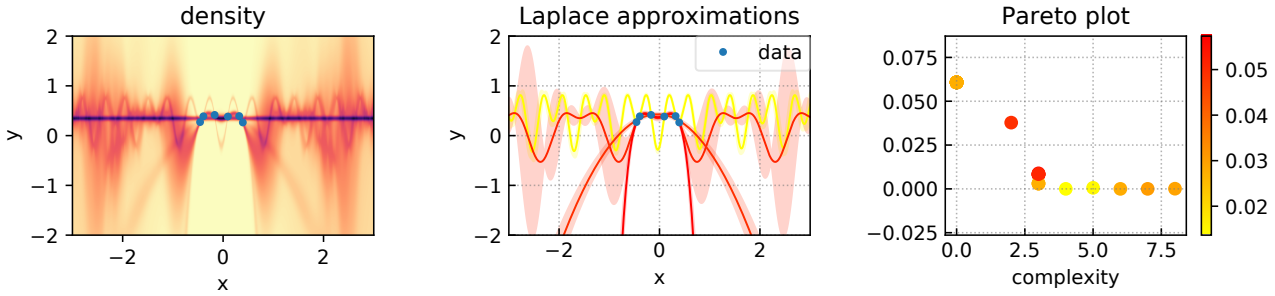
$$y = 0.8 \cos x + \epsilon \,, \qquad \epsilon \sim \mathcal{N}(0, 0.01^2) \tag{E1}$$

$$y = 0.8 \cos x - 0.4 + \epsilon \,, \qquad \epsilon \sim \mathcal{N}(0, 0.03^2) \,. \tag{E2}$$

For each dataset 6 x values were uniformly sampled from $[-1, 1]$. We calculated 31 plausible equations for each dataset with different

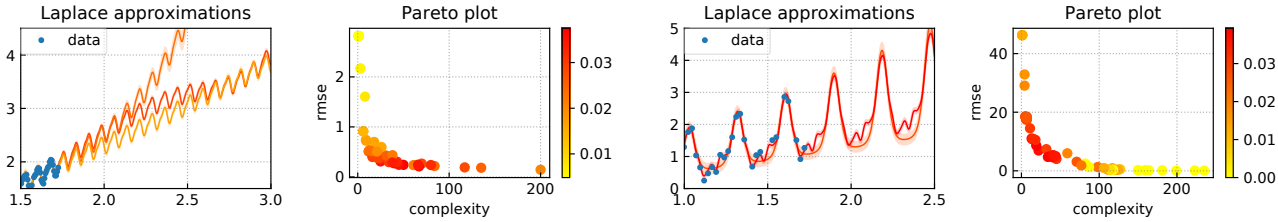


(a) toy example E1



(b) toy example E2

Figure 1: Mixture of Laplace approximations for the illustrative toy examples E1 and E2 in row 1 and 2 respectively. The left panel shows the predictive distribution given by equation 10. The panel in the middle shows individual local Laplace approximations with 2σ standard deviation indicated by the shaded area. The color scheme is chosen such that it is aligned with the pareto plot on the right side. It shows root mean squared error over the complexity of each equation. The color indicates the weighting of the mixture coefficients given by its normalized marginal likelihood. Yellow indicates low weight, and red indicates high weight.



(a) Mauna dataset (b) Airline dataset

Figure 2: Local Laplace approximations with 2σ standard deviation indicated by the shaded area for the Mauna and Airline dataset. The color scheme is chosen such that it is aligned with the pareto plot. It shows root mean squared error (rmse) on the train dataset over the complexity of each equation. The color indicates the weighting of the mixture coefficients. Yellow indicates low weight, and red indicates high weight.

complexities with the iEQL. The right panel of figure 1 shows the pareto plot. The iEQL found equations with complexities in the range of $[0, 8]$ for both toy examples. Their color indicates the weighting by the mixture coefficients π_k. Yellow indicates a small weighting and red indicates a strong weighting. The left panel shows the density distribution of the MoLA for all found equations and the middle panel shows the local Laplace approximation for four hand-picked equations. Their analytic expressions are listed in table 1 along with their mixture coefficients π_k, root mean square error (rmse) and complexity.

The first two equations of toy example E1 are the two dominant modes given by a cos and a parabola equation. They can be clearly identified in the density distribution in the left panel of figure 1. The two other equations are more accurate, but also more complex. Their weighting coefficient is smaller and thus they are hard to

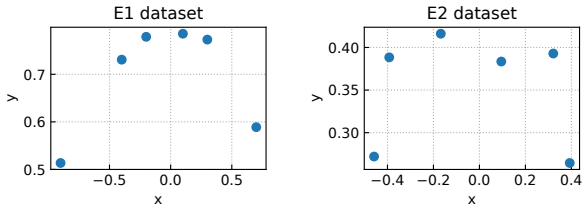


Figure 3: Ambiguous datasets given by equation E1 (left) and equation E2 (right)

identify in the density plot, but their local Laplace approximation is shown in the middle panel.

The first three equations of toy example E2 are the most dominant modes given by a fourth order polynomial, a squared cos and a parabola equation in descending order. The last hand-picked equation is orders of magnitude more accurate yet its mixture coefficient is 4 times smaller compared to the dominant modes. This can be related to larger Eigenvalues of its curvature matrix, meaning less flexibility in parameter space. Its local Laplace approximation is the yellow, high frequency line shown in the middle panel of figure 1.

4.2 Real world time series

In the following section, we investigate two real world time series datasets [1] , which have been studied in the course of structured uncertainty estimates by the *automatic statistician* [5, 20, 34]. We present similar structured uncertainty estimates and provide explainable, analytic expressions instead of nonparametric Gaussian processes for

Mauna Loa atmospheric CO_2 concentration (Mauna) recorded at the Mauna Loa observatory

Airline passenger data (Airline) of monthly totals of international airline passengers.

We focus on extrapolation to examine our method's ability to discover the underlying structured uncertainty. We calculated 51 plausible equations for each dataset with the iEQL. The found equations can capture the underlying structure of the two datasets. The complexity of the found equations lie within $[1 - 235]$ parameters. The mixture coefficients reliably prefer accurate and simple equations as shown in the pareto plots in figure 2. Figure 4 shows the predictive distribution of the MoLa in the left panel and the middle panel shows a cutout area. The predictive distribution clearly indicates that the predictions of the equations diverge in the extrapolation area, as expected. The right panel shows the approximation of the predictive distribution for fast prediction. It captures the underlying structure of the dataset and provides calibrated uncertainty estimates. This is in contrast to the local Laplace approximations, which are known to underestimate the uncertainty as shown in figure 2 for three hand-picked equations for the Mauna dataset and two hand-picked equations for the Airline dataset. Their corresponding mathematical expressions are shown in table 2. We found that our method estimates the correct dominating frequency in all selected equations. In the Mauna dataset the predictions deviate in the extrapolation region due to their differences in structure, which is an important motivation to capture *global uncertainty* with a mixture model. Especially, the second equation models the growth of the data with an exponentially growing contribution e^{ax_1}, whereas the first equation uses a fourth order polynomial and the third equation uses a parabola. In the Airline dataset the structure of the second equation is very simple and does not capture higher frequencies. This leads to a larger uncertainty of its local Laplace approximation.

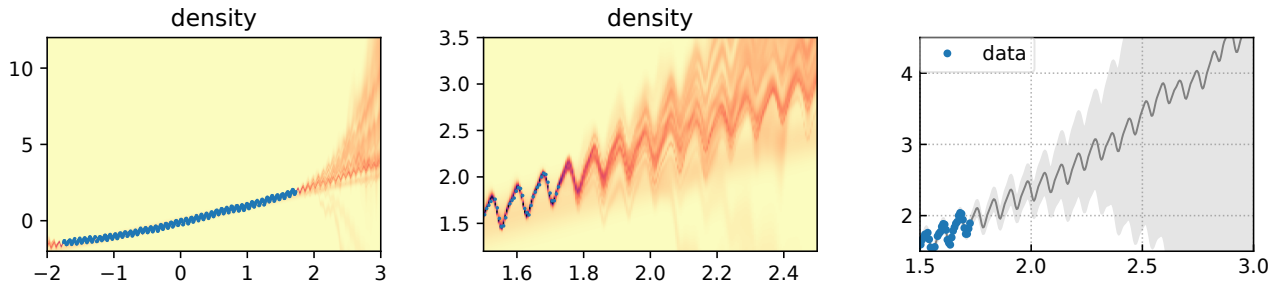
In this section, we showed that our method captures the underlying structure of the datasets and provides structured uncertainty estimates. The individual local Laplace approximations underestimate the global uncertainty. Our approximation for fast prediction provides reliable structured uncertainty, yet this has to be applied with caution since the conditional mean of a multimodal distribution can lead to a poor representation of the data.

5 RELATED WORK

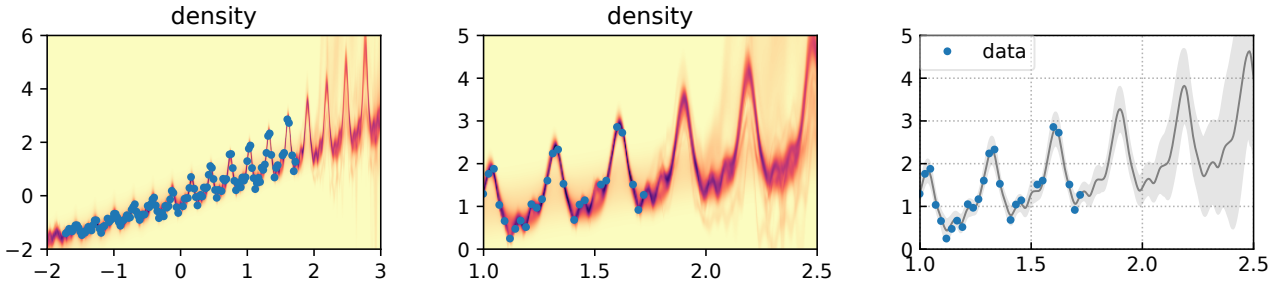
Equation learning. is a topic of growing interest in machine learning. Zaremba et al. [38] present a tree search guided search with n-gram model or recurrent neural networks [15] used Bayesian optimization to search for equations in latent space with a prior on mathematical constraints. Lample and Charton [17] use seq2seq transformers to solve mathematical integration and ODEs. Inspired by physics, Udrescu and Tegmark [35] exploit symmetries and separability in the dataset to enhance the search for equations. More recent publications use gradient information about the learned expression during training. This was first addressed in a reinforcement learning formulation by Petersen [29] via a risk-seeking policy gradients. A different, yet powerful method are equation learning (EQL) neural networks [26, 31, 36]. They represent a complex equation within their architecture, with different kinds of activation functions (e.g. $(\cos, \sin, \log, e, *, /, \dots)$) in each hidden layer. During training irrelevant parts are omitted and it converges to a sparse representation that is the wanted equation itself. Kim et al. [11] integrated it within other deep learning frameworks. Long et al. [21] applied it to differential equations. Lin et al. [19] used it to obtain analytical expressions of classical free energy functionals. Symbolic regression is commonly addressed with genetic programming and evolutionary algorithms [18, 27, 33]. It has been applied to automate the discovery of natural laws [4, 32]. All those methods provide several plausible equations of different complexity. A suitable equation is then chosen by a predefined selection criteria or by the user itself. We propose to combine the set of plausible equations to be consent about the equations *local* and *global* uncertainty.

Automatic statistician: Structured and explainable uncertainty has been studied with Gaussian processes in the context of kernel learning [5, 20, 34, 37]. Especially, the *automatic statistician* aims to learn an explainable structure of base kernels to describe high-level properties like smoothness, trends, periodicity and change points. Learning such structural forms of the kernel also enables for long-range extrapolation. We achieve similar statistical descriptions, yet with explainable equations instead of nonparametric, black box Gaussian processes.

[1]The datasets are downloaded from https://github.com/ssydasheng/Neural-Kernel-Network



(a) Mauna dataset



(b) Airline dataset

Figure 4: Mixture of Laplace approximations for the Mauna and Airline dataset in the first and second row respectively. The left panel shows the predictive distribution given by equation 10. The panel in the middle shows a cutout of the predictive distribution. The panel on the right shows the approximation of the predictive distribution of the MoLA given by equation 13. The shaded area indicates 2σ standard deviation.

Table 2: Hand-picked equations for real world datasets Mauna and Airline with root mean squared error (rmse) and mixture coefficients π_k on the train dataset. Their local Laplace approximations are shown in the middle panel of figure 4.

dataset	π_k	rmse	equation
Mauna	0.029	0.4	$y_0 = 0.89x_1 + 0.42\,(0.52x_1 + 0.22)^2 - 0.15\cos(82.73x_1 - 3.49)$ $+0.23\left(-0.35x_1 + 0.85\,(-0.27x_1 + 0.66\cos(3.39x_1 + 5.09) - 0.28)^2 - 0.49\right)^2$ $-0.1e^{0.41\cos(165.39x_1 - 5.67)} - 0.07$
	0.024	0.51	$y_0 = 0.8x_1 - 0.36\,(0.32 - 0.49\cos(82.69x_1 + 0.31))^2 + 0.09\cos(82.8x_1 - 1.18)$ $+0.01e^{2.42x_1} - 0.43e^{-1.9(-0.56x_1 - 0.43)^2} + 0.28$
	0.017	0.73	$y_0 = 0.83x_1 + 0.66\,(-0.41x_1 - 0.29)^2 + 0.29\,(0.69\cos(82.73x_1 - 14.03) + 0.08)^2$ $+0.13e^{-0.99\cos(82.71x_1 + 2.99)} - 0.4$
Airline	0.039	8.7	$y_0 = 0.41x_1\,(0.23x_1 + 0.28) + 0.61x_1$ $+0.4\,(0.86\,(-0.47\cos(65.09x_1 - 1.95) - 0.52)\,(0.77\cos(65.09x_1 - 1.95) + 0.09)$ $+0.98) \cdot (1.53\cos(21.78x_1 - 0.58) + 0.74\cos(22.06x_1 + 3.28) + 0.04)$ $+0.21e^{0.38x_1 - 1.92\cos(21.78x_1 - 0.58)} + 0.18e^{-2.6(1.02 - 1.92x_1)^2} - 0.63$
	0.027	18	$y_0 = 0.75x_1 + 0.1e^{0.64x_1 - 2.0\cos(21.75x_1 - 6.67)} - 0.28$

Uncertainty estimation for equations: Recent insights in uncertainty estimation with Laplace approximations for neural networks can also be applied to uncertainty estimation for equations, without the disadvantage of huge parameter spaces. The Laplace approximation for neural networks has been first introduced by MacKay [22]. It requires to invert the full Hessian, which is huge for modern neural networks. With recent developments in Hessian approximation [1, 24, 25] the Laplace approximation became accessible to modern neural networks [3, 14, 30]. Foong et al. [7] empirically show that the linearized Laplace approximation leads to better calibrated

uncertainty estimates of simple neural networks than without linearization. We experienced similar effects with equations instead of neural networks. Immer et al. [9] propose to apply the Laplace approximation to the local linearization of the neural network justifying the ggn approximation. But, the Laplace approximation is prone to underestimate the uncertainty.

Lakshminarayanan et al. [16] present a simple, yet state-of-the-art, method for uncertainty estimation in deep learning combining several independently trained neural networks in a deep ensemble. Motivated by their results we propose to use a mixture of Laplace approximations (MoLA,Eschenhagen et al. [6]) for a set of plausible equations.

6 CONCLUSION

We introduced uncertainty in equation learning for any set of plausible equations found with an equation learner. We identified two components of uncertainty: *global uncertainty* given by the differences in structure of each equation and *local uncertainty* given by parametric uncertainty within one family of equations. We introduced a mixture of Laplace approximations to capture *global uncertainty* of several plausible equations. Each mixture component captures a different equation structure and the Laplace approximations capture local, parametric uncertainty within one family of equations. For computationally fast prediction we proposed to match the first two moments of the mixture of Laplace approximations.

ACKNOWLEDGMENTS

Georg Martius and Philipp Hennig are members of the Machine Learning Cluster of Excellence, EXC number 2064/1 – Project number 390727645. We acknowledge the support from the German Federal Ministry of Education and Research (BMBF) through the Tübingen AI Center (FKZ: 01IS18039B).

REFERENCES

[1] Aleksandar Botev, Hippolyt Ritter, and David Barber. 2017. Practical Gauss-Newton optimisation for deep learning. In *International Conference on Machine Learning*. PMLR, 557–565.
[2] Felix Dangel, Frederik Kunstner, and Philipp Hennig. 2020. BackPACK: Packing more into Backprop. In *International Conference on Learning Representations*.
[3] Erik Daxberger, Eric Nalisnick, James Urquhart Allingham, Javier Antorán, and José Miguel Hernández-Lobato. 2020. Expressive yet Tractable Bayesian Deep Learning via Subnetwork Inference. *arXiv preprint arXiv:2010.14689* (2020).
[4] Renáta Dubčáková. 2011. Eureqa: software review. *Genetic Programming and Evolvable Machines* 12, 2 (01 Jun 2011), 173–178.
[5] David Duvenaud, James Lloyd, Roger Grosse, Joshua Tenenbaum, and Ghahramani Zoubin. 2013. Structure Discovery in Nonparametric Regression through Compositional Kernel Search. In *Proceedings of the 30th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 28)*, Sanjoy Dasgupta and David McAllester (Eds.). PMLR, Atlanta, Georgia, USA, 1166–1174.
[6] Runa Eschenhagen, Erik Daxberger, Philipp Hennig, and Agustinus Kristiadi. 2021. Mixtures of Laplace Approximations for Improved Post-Hoc Uncertainty in Deep Learning. *CoRR* abs/2111.03577 (2021). arXiv:2111.03577 https://arxiv.org/abs/2111.03577
[7] Andrew YK Foong, Yingzhen Li, José Miguel Hernández-Lobato, and Richard E Turner. 2019. 'In-Between'Uncertainty in Bayesian Neural Networks. *arXiv preprint arXiv:1906.11537* (2019).
[8] Carlos E. García, David M. Prett, and Manfred Morari. 1989. Model predictive control: Theory and practice—A survey. *Automatica* 25, 3 (1989), 335 – 348.
[9] Alexander Immer, Maciej Korzepa, and Matthias Bauer. 2021. Improving predictions of Bayesian neural nets via local linearization. In *International Conference on Artificial Intelligence and Statistics*. PMLR, 703–711.
[10] Pavel Izmailov, Dmitrii Podoprikhin, Timur Garipov, Dmitry Vetrov, and Andrew Gordon Wilson. 2018. Averaging weights leads to wider optima and better generalization. *arXiv preprint arXiv:1803.05407* (2018).
[11] Samuel Kim, Peter Y Lu, Srijon Mukherjee, Michael Gilbert, Li Jing, Vladimir Čeperić, and Marin Soljačić. 2020. Integration of neural network-based symbolic regression in deep learning for scientific discovery. *IEEE Transactions on Neural Networks and Learning Systems* (2020).
[12] Diederik P Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. international conference on learning representations (2015).
[13] Agustinus Kristiadi, Matthias Hein, and Philipp Hennig. 2020. Being Bayesian, Even Just a Bit, Fixes Overconfidence in ReLU Networks. In *Proceedings of the 37th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 119)*, Hal Daumé III and Aarti Singh (Eds.). PMLR, 5436–5446.
[14] Agustinus Kristiadi, Matthias Hein, and Philipp Hennig. 2020. Learnable Uncertainty under Laplace Approximations. *arXiv preprint arXiv:2010.02720* (2020).
[15] Matt J Kusner, Brooks Paige, and José Miguel Hernández-Lobato. 2017. Grammar variational autoencoder. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org, 1945–1954.
[16] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. 2016. Simple and scalable predictive uncertainty estimation using deep ensembles. *arXiv preprint arXiv:1612.01474* (2016).
[17] Guillaume Lample and François Charton. 2020. Deep Learning For Symbolic Mathematics. In *International Conference on Learning Representations*.
[18] William B Langdon, Riccardo Poli, Nicholas F McPhee, and John R Koza. 2008. Genetic programming: An introduction and tutorial, with a survey of techniques and applications. In *Computational intelligence: A compendium*. Springer, 927–1028.
[19] Shang-Chun Lin, Georg Martius, and Martin Oettel. 2020. Analytical classical density functionals from an equation learning network. *The Journal of Chemical Physics* 152, 2 (2020), 021102.
[20] James Lloyd, David Duvenaud, Roger Grosse, Joshua Tenenbaum, and Zoubin Ghahramani. [n. d.]. Automatic Construction and Natural-Language Description of Nonparametric Regression Models. 28 ([n. d.]).
[21] Zichao Long, Yiping Lu, and Bin Dong. 2019. PDE-Net 2.0: Learning PDEs from data with a numeric-symbolic hybrid deep network. *Journal of Comp. Physics* 399 (2019), 108925.
[22] David JC MacKay. 1992. A practical Bayesian framework for backpropagation networks. *Neural computation* 4, 3 (1992), 448–472.
[23] Wesley J Maddox, Pavel Izmailov, Timur Garipov, Dmitry P Vetrov, and Andrew Gordon Wilson. 2019. A simple baseline for Bayesian uncertainty in deep learning. *Advances in Neural Information Processing Systems* 32 (2019), 13153–13164.
[24] James Martens. 2020. New Insights and Perspectives on the Natural Gradient Method. *Journal of Machine Learning Research* 21, 146 (2020), 1–76.
[25] James Martens and Roger Grosse. 2015. Optimizing Neural Networks with Kronecker-factored Approximate Curvature. In *Proceedings of the 32nd International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 37)*, Francis Bach and David Blei (Eds.). PMLR, Lille, France, 2408–2417.
[26] Georg Martius and Christoph H Lampert. 2016. Extrapolation and learning equations. *arXiv preprint arXiv:1610.02995* (2016).
[27] Randall K. McRee. 2010. Symbolic Regression Using Nearest Neighbor Indexing. In *Proceedings of the 12th Annual Conference Companion on Genetic and Evolutionary Computation* (Portland, Oregon, USA) *(GECCO '10)*. ACM, New York, NY, USA, 1983–1990.
[28] Tim Pearce, Felix Leibfried, and Alexandra Brintrup. 2020. Uncertainty in neural networks: Approximately Bayesian ensembling. In *International conference on artificial intelligence and statistics*. PMLR, 234–244.
[29] Brenden K Petersen. 2019. Deep symbolic regression: Recovering mathematical expressions from data via policy gradients. *arXiv preprint arXiv:1912.04871* (2019).
[30] Hippolyt Ritter, Aleksandar Botev, and David Barber. 2018. A scalable Laplace approximation for neural networks. In *6th International Conference on Learning Representations, ICLR 2018-Conference Track Proceedings*, Vol. 6. International Conference on Representation Learning.
[31] Subham S. Sahoo, Christoph H. Lampert, and Georg Martius. 2018. Learning Equations for Extrapolation and Control. In *International Conference on Machine Learning, ICML 2018*, Vol. 80. PMLR, 4442–4450.
[32] Michael Schmidt and Hod Lipson. 2009. Distilling Free-Form Natural Laws from Experimental Data. *Science* 324, 5923 (2009), 81–85.
[33] Dominic P Searson, David E Leahy, and Mark J Willis. 2010. GPTIPS: an open source genetic programming toolbox for multigene symbolic regression. In *International Multiconference of Engineers and Computer scientists*, Vol. 1. IMECS Hong Kong, 77–80.
[34] Shengyang Sun, Guodong Zhang, Chaoqi Wang, Wenyuan Zeng, Jiaman Li, and Roger Grosse. 2018. Differentiable compositional kernel learning for Gaussian processes. In *International Conference on Machine Learning*. PMLR, 4828–4837.
[35] Silviu-Marian Udrescu and Max Tegmark. 2020. AI Feynman: A physics-inspired method for symbolic regression. *Science Advances* 6, 16 (2020), eaay2631.

[36] Matthias Werner, Andrej Junginger, Philipp Hennig, and Georg Martius. 2021. Informed Equation Learning. arXiv:2105.06331 [cs.LG]
[37] Andrew Gordon Wilson, Elad Gilboa, John P Cunningham, and Arye Nehorai. 2014. Fast Kernel Learning for Multidimensional Pattern Extrapolation.. In *NIPS*. 3626–3634.
[38] Wojciech Zaremba, Karol Kurach, and Rob Fergus. 2014. Learning to discover efficient mathematical identities. In *Advances in Neural Information Processing Systems*. 1278.

A APPENDIX

A.1 iEQL Training

All experiments are performed using an iEQL architecture [36] with three hidden layers for the real world datasets and two hidden layers for the toy datasets. Each hidden layer has $\{\cos, \exp, x^2, *\}$ as atomic units. Each atomic unit is applied four times in each layer. To account for the high frequency in the real world dataset we use ten $\cos(f\,x)$ units with frequency $f = 80$ instead of four cos units in each hidden layer. This mainly affects the initialization of the iEQL leading to a larger spectrum of frequencies due to the random initialization of the weight matrices. We prohibit combinations of cos(cos), exp(exp).

We do not provide any domain expert knowledge and choose the domain specific complexity factors uniformly. For the given datasets, penalty epochs are not necessary. We use the proposed optimizer setting with Adam and a learning rate $\alpha = 0.001$ without mini batches because of the high frequency of the data. The toy examples are trained for $T_1 = 60000$ iterations without regularization and for $T_2 = 100000$ iteration with regularization. The real world dataset are trained for $T_1 = 200000$ iterations without regularization and for $T_2 = 600000$ iteration with regularization.

After convergence, we fine-tune the found equation with 40 steps with an lbfgs optimizer with $\alpha = 1$ and a *strong wolfe* line searchi, since Adam is not guaranteed to converge.

In order to capture plausible equations of different complexity and accuracy, we train the iEQL with different regularization strengths $\lambda = 10^k$ where k is in the range from -3.0 to 0.0 with 31 equally spaced steps for the toy examples and the range from -5 to 0.0 with 51 equally spaced steps for the real world datasets.

Each training for a single regularization strength was executed on a single CPU. A training lasts 9096 ± 2295 s for the Airline dataset, 11406 ± 1301 s for the Mauna dataset, 1578 ± 288 s for the toy dataset E1 and 1645 ± 180 s for the toy dataset E2.

Real world dataset preparation: We normalize input and output of the real world datasets for training. The datasets are split into 90% training and 10% testing.