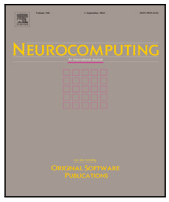
Neurocomputing 585 (2024) 127645

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom



Normalizing flow based uncertainty estimation for deep regression analysis

Baobing Zhang [a], Wanxin Sui [a], Zhengwen Huang [a], Maozhen Li [a,*], Man Qi [b]

[a] Department of Electronic and Electrical Engineering, Brunel University London Kingston Lane, Uxbridge, Middlesex, UB8 3PH, UK
[b] School of Engineering, Technology and Design, Canterbury Christ Church University, Canterbury, CT1 1QU, UK

ARTICLE INFO

Communicated by N. Zeng

Keywords:
Regression
Predictive uncertainty
Normalizing flow
Probabilistic modeling
Adversarial robustness
Calibration

ABSTRACT

Uncertainty estimation is a critical component of building safe and reliable machine learning models. Accurate estimation of uncertainties is essential for identifying and mitigating potential risks and ensuring that machine learning systems operate reliably in real-world scenarios. Various approaches, such as ensemble and Bayesian neural networks have been developed by sampling probability predictions from submodels, which is computationally expensive. At present, these techniques are incapable of precisely delineating the boundary separating in-distribution (ID) and out-of-distribution (OOD) data. To fill up this research gap, this paper presents a normalizing flow based framework to directly predict parameters of prior distributions over the probability with a neural network, the proposed model is able to effectively differentiate between ID and OOD data in regression problems. The posterior distributions learned by the model precisely represent uncertainties for OOD data based solely on ID data, without the need for OOD data during training. This approach has shown promising results in a number of applications, including image depth estimation and image adversarial attacks.

1. Introduction

The exceptional performance of neural networks across a range of tasks has led to their widespread adoption in numerous fields, including computer vision [1,2] and natural language processing [3,4]. Machine learning(ML) models can give a particularly good result in most cases, but occasionally give a particularly bad result, but this particularly bad result is absolutely unacceptable in many life critical situations such as aerospace, biomedical, autonomous-driving. If a ML model produces unsatisfactory outcomes with low confidence, human intervention can be employed to rectify the error. This approach ensures that the model can be utilized safely across a broader range of domains. Uncertainty estimation in neural networks has therefore become an active area of research, with a range of methods [5–13] having been proposed in recent years.

Aleatoric and epistemic uncertainty are two types of uncertainty that are encountered in ML and other fields that involve predictions and modeling. Aleatoric uncertainty is uncertainty that arises as a consequence of the intrinsic randomness or variability in the data. As an illustration, if someone intends to forecast the result of a coin flip, there is inherent randomness in the process, and even if you have perfect knowledge of the physical conditions, you cannot predict the outcome with complete certainty. In ML, aleatoric uncertainty arises from sources such as measurement noise, natural variation in the data, or incomplete or inaccurate data. On the contrary, epistemic uncertainty pertains to uncertainty that emerges as a result of a scarcity of knowledge or comprehension regarding the system that is being modeled. If you are trying to predict the outcome of a medical test based on a patient's symptoms, there may be factors that are unknown or not fully understood that affect the outcome. In ML, epistemic uncertainty arises from sources such as limited training data, model misspecification, or incomplete knowledge of the underlying mechanisms that generate the data. Both types of uncertainty can affect the reliability and accuracy of predictions, and understanding the sources of uncertainty can help in developing more robust and accurate ML models.

When it comes to real-world deployment in safety-critical domains, regression models face high accurate requirements. The typical approach of training a deep neural networks (DNNs) to produce a predicted regression target $\hat{y} = f(x)$ is inadequate in capturing any level of uncertainty in the predictions $\hat{y}$. This deficiency renders the model incapable of detecting OOD input x, which are not part of its training data. Since the accuracy of DNNs' predictions typically declines significantly on OOD input [14,15], the potential consequences could be disastrous. Therefore, various techniques have been developed to train uncertainty-aware DNNs models [5,9,16–21] to explicitly estimate uncertainty in the predictions.

* Corresponding author.
E-mail addresses: Baobing.Zhang2@brunel.ac.uk (B. Zhang), cynthia.sui@brunel.ac.uk (W. Sui), zhengwen.huang@brunel.ac.uk (Z. Huang), maozhen.li@brunel.ac.uk (M. Li), man.qi@canterbury.ac.uk (M. Qi).

https://doi.org/10.1016/j.neucom.2024.127645
Received 8 January 2024; Received in revised form 4 March 2024; Accepted 29 March 2024
Available online 4 April 2024

Accurate and reliable uncertainty estimates are essential. Otherwise, the model's occasional overconfidence may lead to overconfident yet incorrect predictions, making uncertainty estimates misleading and potentially increasing the model's unsuitability for safety-critical deployment. To be effective, uncertainty estimates must be *well calibrated* and aligned with prediction errors [22], even in situations involving distribution shifts encountered during practical deployment [23–25]. For example, a autopilot model trained on data from a large urban street in 2020 should generate well-calibrated predictions for cars from both urban and rural areas in 2023. Although uncertainty calibration and general DNNs robustness [26] have been evaluated for classification tasks [27,28] under distribution shifts, this important issue remains understudied for regression.

One frequently used approach is to leverage Bayesian Neural Networks (BNNs). BNNs use Bayes' rule to create a model of the posterior distribution over the weights of a neural network by treating them as random variables with a prior distribution. Through the use of Bayes' rule to update the posterior distribution during training, BNNs can capture the variation in the network's weights and measure the uncertainty in their forecasts [5,29]. However, BNNs face several limitations, including lack of interpretability, limited scalability, computational complexity and the choice of priors makes this method difficult to apply on a large scale of data set. Ensembling [30] involves training multiple models on the same dataset with different initializations, which can be time-consuming and require significant computing resources. If the models in the ensemble are too complex or too similar, they may overfit the training data and not generalize well to new data. Dropout [31] is a regularization technique that randomly drops out units during training to reduce overfitting. However, the dropout rate is typically set to a fixed value during training and does not change during testing. This can result in unreliable uncertainty estimates, especially for OOD samples. The multiple models in an ensemble or the random units dropped out during training can make it difficult to interpret the model's predictions and uncertainty estimates. This can be a problem in domains where interpretability is important, such as healthcare or finance. In practical applications, most ML models do not have the ability to perceive uncertainty, and modeling uncertainty in the face of specific practical problems still remains challenging.

The normalizing flow [32,33] model has attracted attention because of its reversibility without loss of original information, and it has proven to have remarkable performance in various applications includes clustering and classification [27,34], and density estimation [35,36]. In this paper, we present a normalizing flow based uncertainty estimation framework (FlowNet) for regression analysis. FlowNet is trained on ID data directly predicting parameters of prior distributions. Through the normalizing flow model, the density of ID samples are aggregated together in a latent space, and when an OOD sample is detected, FlowNet will separate ID from the OOD sample in the latent space, then the model will give a low confidence. The outcomes achieved by FlowNet in detecting Out-of-Distribution (OOD) and accurately estimating uncertainty during dataset shifts are considered state-of-the-art. Notably, FlowNet accomplishes this without requiring OOD samples during training or relying on costly sampling techniques for uncertainty estimation during testing.

The contributions of the paper are as follows :

1. It presents a normalizing flow based uncertainty estimation framework that is capable of accurately detecting uncertainty without requiring additional OOD training data.
2. It introduces a novel uncertainty bounding mechanism, which is capable of effectively identifying the data whose distribution is out of the target distribution in a latent space
3. A thorough evaluation of epistemic uncertainty is performed on both standard benchmark and intricate visual regression tasks, with a comparison to traditional neural network uncertainty estimation methods.
4. The evaluation of both robustness and calibration is performed using OOD and adversarially perturbed test data.

The remaining sections of this paper are structured in the following manner. Section 2 gives a review of related work. Section 3 introduces the necessary prerequisite and details the formulation of learning process. Section 4 presents the experimental results and data analysis. Section 5 concludes this paper and points out future direction.

2. Related work

Uncertainty estimation is an essential aspect of ML. In the early days of ML, uncertainty was mostly ignored. However, over time, it became clear that uncertainty was a crucial factor in many applications. One of the early methods for estimating uncertainty in ML was Monte Carlo Dropout (MC Dropout), proposed by Gal and Ghahramani in 2016 [10]. MC Dropout is a technique for estimating uncertainty by repeatedly sampling from a trained model with dropout enabled. Afterwards, a series of uncertainty estimation methods based on dropout have been proposed [10,37–39], but methods based on dropout can be computationally expensive. Another popular method for uncertainty estimation is Bayesian neural networks (BNNs) [5–7]. BNNs are based on Bayesian statistics, which provides a way to estimate the uncertainty of a model's predictions by treating the model parameters as random variables. BNNs provide a principled way to estimate uncertainty, but they can be difficult to train and may require large amounts of computational resources. Other commonly used methods for uncertainty estimation include ensemble methods [40], which combine the predictions of multiple models to estimate uncertainty, and deep ensembles [9], which use ensembles of deep neural networks to estimate uncertainty. Ensemble methods are straightforward to implement and can be very effective, but they may require more memory and computation.

Recently, a novel class of models has been created with the aim of forecasting the parameters of a prior distribution on sample probability predictions, while taking into consideration diverse forms of uncertainty [41–45]. Nevertheless, these approaches come with limitations. Prior Networks [42,43] employ OOD samples during training to learn these parameters and establish distinct target values for ID and OOD data. There are several issues with this approach: (1) Anticipating knowledge of Out-of-Distribution (OOD) data during training is unrealistic as such samples are unlikely to be observed in practical scenarios. (2) Providing a specific set of out-of-distribution (OOD) samples cannot effectively distinguish between in-distribution (ID) and OOD data. This is because any data that does not come from the original data distribution is considered OOD. As a result, characterizing an infinitely large OOD distribution with a finite data set is impossible. (3) The predicted prior distribution parameters are not restricted to any specific values, particularly for new OOD samples that were not included in the training dataset. Furthermore, the total number of fabricated pseudo-observations over the input domain may surpass the number of actual observations, resulting in unwanted behaviors and conferring unwarranted epistemic certainty to OOD data that was not encountered during training. Based on [27], it appears that depriving models of explicit OOD data during training using these techniques leads to unfavorable results.

The evidential model [45] learns a higher-order distribution based on a lower-order distribution. Statistically analyze the numerical characteristics of higher-order distributions at test time to give uncertainty predictions. It learns the training data in the latent space, which is infinite and unbounded, which makes it difficult to penalize evidence everywhere. Compared with it, FlowNet leverages normalizing flows to latch on knowledge of a distribution across the conjugate prior distribution of Gaussian distribution parameters in a latent space, which more precisely recognizes the ID and OOD data, due to the data is mapped to a fixed and finite probability density region. As a result, data mapping within this region is considered ID data, and data outside the specific

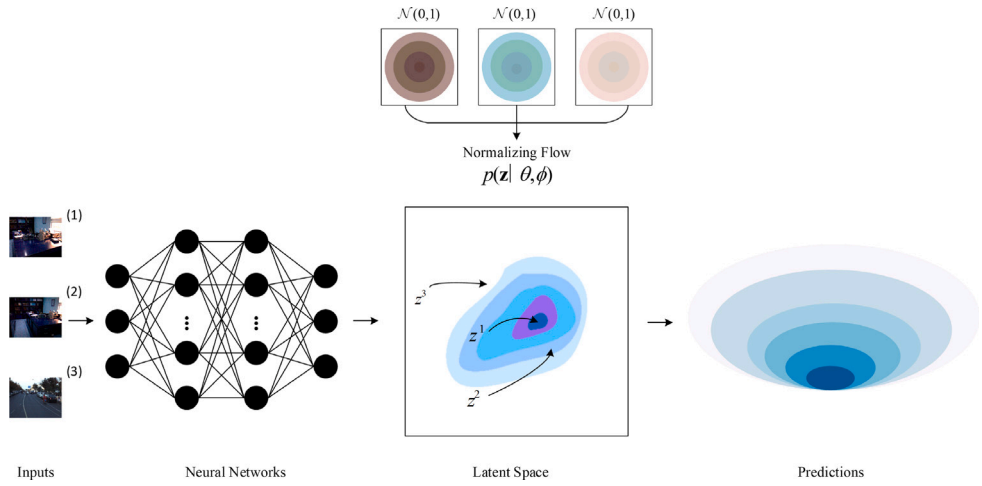


Fig. 1. Three input examples, (1), (2), and (3) are fed into the encoding neural network to obtain their corresponding latent space coordinates z. Higher probability mass is represented by darker shading. z^1 and z^2 falling in the high density area are considered ID data, while z^3 falling outside the density area is considered OOD data. The normalizing flow component learns normalized density functions $p(\mathbf{z} \mid \theta, \phi)$, this density is further used to parameterize the Normal Inverse-Gamma distribution then give the prediction and uncertainty estimation.

region is considered as OOD data. There is boundary betwixt ID and OOD samples in the latent space. At the same time, FlowNet does not require additional OOD data at training time. Fig. 1 makes clear the flowchart.

3. FlowNet for regression

3.1. Prerequisite

In ML, regression refers to a supervised learning technique used to model the relationship between a dependent variable and one or more independent variables. The goal of regression is to find a function that can predict the value of the dependent variable based on the values of the independent variables. Differ from classification, the output of regression model is real-value attributes for the data instances, instead of the predefined classes that the data belong to. The quality of the regression model is typically evaluated based on metrics such as mean squared error, root mean squared error, and others. Formally, given a dataset D, which is made up of N pair training examples, it is expressed as $D = \{x_i, y_i\}_{i=1}^{N}$. The optimization process is achieved by adjusting the values of the weights w in order to learn a functional f.

$$\min_{\boldsymbol{w}} J(\boldsymbol{w}); \quad J(\boldsymbol{w}) = \frac{1}{N}\sum_{i=1}^{N} \mathcal{L}_i(\boldsymbol{w}) \tag{1}$$

where $\mathcal{L}_i(\cdot)$ is the loss function. Sum of squared errors is the commonly used objective function, $\mathcal{L}_i(\boldsymbol{w}) = \frac{1}{2}\left\|y_i - f\left(\boldsymbol{x}_i; \boldsymbol{w}\right)\right\|^2$, it typically optimizes a model by rewarding correct predictions and penalizing incorrect ones. However, this cannot fit the potential noise and uncertainty estimates when the test data is completely different from the training data.

From the perspective of probabilities, it allows predictions to be made in face of uncertainty. Assume the targets y_i were drawn i.i.d. from Gaussian distribution with mean and variance parameters $\boldsymbol{\theta} = \left(\mu, \sigma^2\right)$. The objective of maximum likelihood estimation (MLE) is to train a model to determine the value of parameter $\boldsymbol{\theta}$ that maximizes the probability of observing the target outputs, y, as given by the function $p\left(y_i \mid \boldsymbol{\theta}\right)$. This is accomplished by minimizing the loss function of negative log likelihood.

$$\mathcal{L}_i(\boldsymbol{w}) = -\log p(y_i \mid \underbrace{\mu, \sigma^2}_{\theta}) = \frac{1}{2}\log\left(2\pi\sigma^2\right) + \frac{\left(y_i - \mu\right)^2}{2\sigma^2} \tag{2}$$

The learned parameter $\boldsymbol{\theta}$ will vary according to different datasets. Uncertainty is then estimated from the numerical properties of learned dataset in statistics. This kind of method, can only model the uncertainty inside the dataset, which is commonly referred to as aleatoric uncertainty, but does not have the ability to estimate the epistemic uncertainty [17]. Implicitly modeling the prior distribution, approaches such as ensemble [9] and dropout [10] have their limitations, as they may sacrifice the estimation of statistics for the sake of S samples. The model can learn hyperparameters of the prior distribution by explicitly placing priors over the likelihood function, which is the approach taken by a group of methods [41–45], without the need for sampling, it is possible to accurately represent both epistemic and aleatoric uncertainty.

3.2. Problem formulation

The problem we are considering involves observed targets, y_i, drawn independent and identically distributed from a Gaussian distribution, aim to estimate the probabilistic values of unknown mean and variance $\left(\mu, \sigma^2\right)$ using a method similar to classic Maximum Likelihood Estimation (MLE) (Section. III.A). To achieve this, we introduce a prior distribution on $\left(\mu, \sigma^2\right)$. If it is assumed observations are sampled from a Gaussian, as described in Section. III.A, we use Inverse-Gamma prior and Gaussian prior for the unknown variance and mean respectively .

$$\begin{aligned} \left(y_1, \ldots, y_N\right) &\sim \mathcal{N}\left(\mu, \sigma^2\right) \\ \sigma^2 \sim \Gamma^{-1}(\alpha, \beta) \quad \mu &\sim \mathcal{N}\left(\gamma, \sigma^2 \upsilon^{-1}\right) \end{aligned} \tag{3}$$

where $\beta > 0, \alpha > 1, \upsilon > 0, \gamma \in \mathbb{R}$, $\boldsymbol{m} = (\beta, \alpha, \upsilon, \gamma)$ and $\Gamma(\cdot)$ is refer to as gamma function. FlowNet will estimate a posterior distribution $q\left(\mu, \sigma^2\right) = p\left(\mu, \sigma^2 \mid y_1, \ldots, y_N\right)$. Assuming that the estimated distribution can be factorized [46], we obtain an approximation for the true posterior $q\left(\mu, \sigma^2\right) = q(\mu)q\left(\sigma^2\right)$. The approximation we use is in the form of the Normal Inverse-Gamma (N-Γ^{-1}) distribution, which is a Gaussian conjugate prior distribution.

$$\begin{aligned} p(\underbrace{\mu, \sigma^2}_{\boldsymbol{\theta},\boldsymbol{\phi}} \mid \underbrace{\beta, \alpha, \upsilon, \gamma}_{\boldsymbol{m}}) = \frac{\beta^\alpha \sqrt{\upsilon}}{\Gamma(\alpha)\sqrt{2\pi\sigma^2}}\left(\frac{1}{\sigma^2}\right)^{\alpha+1} \\ \exp\left\{-\frac{2\beta + \upsilon(\gamma - \mu)^2}{2\sigma^2}\right\}. \end{aligned} \tag{4}$$

The FlowNet model's parameterization is essential and relies on two main components. An encoder neural network f_θ is the first part of FlowNet, the inputs $\mathbf{X}^{(i)}$ is then mapped into a high-dimensional feature space. The second component is a normalizing flow model parameterized by ϕ, is used to learn a normalized sample density on

this latent space. According to [47], one way to understand the parameters associated with the corresponding conjugate prior distribution is through the concept "pseudo observations". Such as, the variance of the N-Γ^{-1} distribution could thought of deriving from α pseudo observations accompany by a number 2υ as sum of squared deviations and with sample mean α. Whereas, the mean is estimated from υ pseudo observations accompany by a number γ as sample mean. Based on the stated perspective, we can define the total pseudo count, denoted by $\boldsymbol{\Phi}$, of the target distribution as summation of all deduced pseudo observations count, which is equal to 2υ plus α. It is important to note that the second part of FlowNet must be a proper normalized density function to make sure that the model's epistemic uncertainty increases, while sample lie out of known distribution. Our approach centers around the core concept of utilizing normalizing flows to parameterize distributions. Normalizing flows [33], such as radial flow [32], RealNVP [48] or MAF [49], offer a flexible yet manageable family of distributions. It is worth noting that empowered with a sufficiently expressive and deep model [50,51], normalizing flows can theoretically model any continuous distribution.

3.3. Estimation of uncertainty

The two types of uncertainty in a prediction can be classified as aleatoric uncertainty, which is also known as statistical or data uncertainty, and epistemic uncertainty, which represents the lack of knowledge in the prediction. By using N-Γ^{-1} distribution, we can calculate the epistemic uncertainty, aleatoric uncertainty and prediction.

$$\underbrace{\mathbb{E}\left[\sigma^2\right]=\frac{\beta}{\alpha-1}}_{\text{aleatoric}},\quad \underbrace{\mathrm{Var}[\mu]=\frac{\beta}{\upsilon(\alpha-1)}}_{\text{epistemic}},\quad \underbrace{\mathbb{E}[\mu]=\gamma}_{\text{prediction}}. \tag{5}$$

According to the nature of the N-Γ^{-1} distribution, we can understand aleatoric uncertainty and epistemic uncertainty as the mean of the variance and the variance of the mean, respectively.

3.4. Learning target distribution

After formalizing the use of N-Γ^{-1} distribution to obtain both epistemic and aleatoric uncertainty, our consequent step is to train a model that outputs outcome hyperparameters of this distribution. To make the learning process clearer, we divide it into two distinct parts. The first part involves obtaining and maximizing model evidence to support for the observation, while the second part involves inflating uncertainty and minimizing evidence when prediction is incorrect. Broadly speaking, the first part involves fitting data to FlowNet, while the second part enforces a prior that removes inaccurate observation and inflates uncertainty.

To maximize the model fit, one can employ the Bayesian probability theory and utilize the marginal likelihood, also known as the model evidence. This quantity represents the probability of observing the data, y_i, given the values of the distribution parameters, $\boldsymbol{m}$, and is obtained by integrating over the possible values of the likelihood parameters, $\boldsymbol{\theta},\boldsymbol{\phi}$:

$$\begin{aligned} p\left(y_i \mid \boldsymbol{m}\right) &= \frac{p\left(y_i \mid \boldsymbol{\theta},\boldsymbol{\phi},\boldsymbol{m}\right) p(\boldsymbol{\theta},\boldsymbol{\phi} \mid \boldsymbol{m})}{p\left(\boldsymbol{\theta},\boldsymbol{\phi} \mid y_i,\boldsymbol{m}\right)} \\ &= \int_{\sigma^2=0}^{\infty}\int_{\mu=-\infty}^{\infty} p\left(y_i \mid \mu,\sigma^2\right) p\left(\mu,\sigma^2 \mid \boldsymbol{m}\right) \mathrm{d}\mu\mathrm{d}\sigma^2 \end{aligned} \tag{6}$$

Evaluating the model evidence is typically a challenging task as it requires integrating over the latent model parameters. However, if we use a N-Γ^{-1} prior for our Gaussian likelihood function, an analytical solution can be obtained:

$$\begin{aligned} p\left(y_i \mid \boldsymbol{m}\right) &= \int_{\boldsymbol{\theta},\boldsymbol{\phi}} p\left(y_i \mid \boldsymbol{\theta},\boldsymbol{\phi}\right) p(\boldsymbol{\theta},\boldsymbol{\phi} \mid \boldsymbol{m})\mathrm{d}(\boldsymbol{\theta},\boldsymbol{\phi}) \\ &= \int_{\sigma^2=0}^{\infty}\int_{\mu=-\infty}^{\infty} p\left(y_i \mid \mu,\sigma^2\right) p\left(\mu,\sigma^2 \mid \boldsymbol{m}\right) \mathrm{d}\mu\mathrm{d}\sigma^2 \\ &= \int_{\sigma^2=0}^{\infty}\int_{\mu=-\infty}^{\infty} p\left(y_i \mid \mu,\sigma^2\right) p\left(\mu,\sigma^2 \mid \gamma,\upsilon,\alpha,\beta\right) \mathrm{d}\mu\mathrm{d}\sigma^2 \\ &= \int_{\sigma^2=0}^{\infty}\int_{\mu=-\infty}^{\infty} \left[\sqrt{\frac{1}{2\pi\sigma^2}}\exp\left\{-\frac{\left(y_i-\mu\right)^2}{2\sigma^2}\right\}\right] \\ &\left[\frac{\beta^\alpha\sqrt{\upsilon}}{\Gamma(\alpha)\sqrt{2\pi\sigma^2}}\left(\frac{1}{\sigma^2}\right)^{\alpha+1}\exp\left\{-\frac{2\beta+\upsilon(\gamma-\mu)^2}{2\sigma^2}\right\}\right]\mathrm{d}\mu\mathrm{d}\sigma^2 \\ &= \int_{\sigma^2=0}^{\infty}\frac{\beta^\alpha\sigma^{-3-2\alpha}}{\sqrt{2\pi}\sqrt{1+1/\upsilon}\Gamma(\alpha)}\exp\left\{-\frac{2\beta+\frac{\upsilon(y_i-\gamma)^2}{1+\upsilon}}{2\sigma^2}\right\}\mathrm{d}\sigma^2 \\ &= \int_{\sigma=0}^{\infty}\frac{\beta^\alpha\sigma^{-3-2\alpha}}{\sqrt{2\pi}\sqrt{1+1/\upsilon}\Gamma(\alpha)}\exp\left\{-\frac{2\beta+\frac{\upsilon(y_i-\gamma)^2}{1+\upsilon}}{2\sigma^2}\right\}2\sigma\mathrm{d}\sigma \\ &= \frac{\Gamma(1/2+\alpha)}{\Gamma(\alpha)}\sqrt{\frac{\upsilon}{\pi}}(2\beta(1+\upsilon))^\alpha \\ &\left(\upsilon\left(y_i-\gamma\right)^2+2\beta(1+\upsilon)\right)^{-\left(\frac{1}{2}+\alpha\right)} \\ p\left(y_i \mid \boldsymbol{m}\right) &= \mathrm{St}\left(y_i;\gamma,\frac{\beta(1+\upsilon)}{\upsilon\alpha},2\alpha\right). \end{aligned} \tag{7}$$

The Student-t distribution with degrees of freedom υ_{St}, scale σ^2_{St} and location μ_{St} is denoted by $\mathrm{St}\left(y;\mu_{\mathrm{St}},\sigma^2_{\mathrm{St}},\upsilon_{St}\right)$, where y represents the input. The negative logarithm of the model evidence is expressed as the loss function $\mathcal{L}_i^{\mathrm{NLL}}(\boldsymbol{w})$.

$$\begin{aligned} \mathcal{L}_i^{\mathrm{NLL}}(\boldsymbol{w}) &= \frac{1}{2}\log\left(\frac{\pi}{\upsilon}\right)-\alpha\log(\Omega)+\left(\alpha+\frac{1}{2}\right)\log \\ &\left(\left(y_i-\gamma\right)^2\upsilon+\Omega\right)+\log\left(\frac{\Gamma(\alpha)}{\Gamma\left(\alpha+\frac{1}{2}\right)}\right) \end{aligned} \tag{8}$$

where $\Omega=2\beta(1+\upsilon)$. By maximizing the model evidence, a neural network can be trained to output parameters of the N-Γ^{-1} distribution that fit the input observations. The loss function $\mathcal{L}_i^{\mathrm{NLL}}(\boldsymbol{w})$ serves as an objective for this training process.

To regularize the training process (penalty on incorrect evidence), a technique is introduced where an incorrect evidence penalty is applied to minimize evidence on incorrect predictions. In the setting of classification, its effectiveness has been proven [44]. For the regression case, a similar minimization involves $KL[p(\boldsymbol{\theta},\boldsymbol{\phi} \mid \boldsymbol{m})\|p(\boldsymbol{\theta},\boldsymbol{\phi} \mid \tilde{\boldsymbol{m}})]$, where $\tilde{m}$ represents the parameter belong to arbitrary N-Γ^{-1} prior with zero evidence . However, the KL between arbitrary N-Γ^{-1} and N-Γ^{-1} with zero evidence prior is undefined, these approaches to regularizing evidential learning are not applicable in regression. An alternative approach is that, by introducing some non-zero evidence (ϵ-evidence) to make the KL finite and defined. However, this would cause hypersensitivity to the selection of the ϵ value, leading to highly unstable training. Therefore, this alternative is not a practical solution. As a result, we employ methods that directly penalize incorrect evidence.

$$\mathcal{L}_i^{\mathrm{R}}(\boldsymbol{w}) = \left|y_i-\mathbb{E}\left[\mu_i\right]\right|\cdot\boldsymbol{\Phi} = \left|y_i-\gamma\right|\cdot(2\upsilon+\alpha) \tag{9}$$

The complete cost function, $\mathcal{L}_i(\boldsymbol{w})$, includes two distinct loss terms that serve to maximize and regularize evidence. A regularization coefficient (λ) is applied to these two terms to appropriately scale their contributions within the total loss.

$$\mathcal{L}_i(\boldsymbol{w}) = \mathcal{L}_i^{\mathrm{NLL}}(\boldsymbol{w})+\lambda\mathcal{L}_i^{\mathrm{R}}(\boldsymbol{w}) \tag{10}$$

Table 1
RMSE and negative log-likelihood (NLL) Benchmark tests summary in statistics. dropout sampling [10], model ensembling [9], evidential regression [45] and our proposed FlowNet. The best results for each dataset and metric are highlighted in bold, with a sample size of 5 for the baseline methods. On almost all datasets, FlowNet surpasses baseline methods in terms of NLL and RMSE performance.

Datasets	RMSE				NLL			
	Dropout	Ensembles	Evidential	FlowNet	Dropout	Ensembles	Evidential	FlowNet
Boston	2.97 ± 0.19	3.28 ± 1.00	3.06 ± 0.16	$\mathbf{2.38 \pm 0.22}$	2.46 ± 0.06	2.41 ± 0.25	$\mathbf{2.35 \pm 0.06}$	$\mathbf{2.24 \pm 0.07}$
Concrete	$\mathbf{5.23 \pm 0.12}$	6.03 ± 0.58	5.85 ± 0.15	5.81 ± 0.19	$\mathbf{3.04 \pm 0.02}$	3.06 ± 0.18	$\mathbf{3.01 \pm 0.02}$	3.09 ± 0.02
Energy	1.66 ± 0.04	2.09 ± 0.29	2.06 ± 0.10	$\mathbf{0.93 \pm 0.17}$	1.99 ± 0.02	1.38 ± 0.22	1.39 ± 0.06	$\mathbf{1.10 \pm 0.09}$
Kin8 nm	0.10 ± 0.00	0.09 ± 0.00	0.09 ± 0.00	$\mathbf{0.05 \pm 0.00}$	-0.95 ± 0.01	-1.20 ± 0.02	-1.24 ± 0.01	$\mathbf{-1.37 \pm 0.03}$
Naval	0.01 ± 0.00	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	-3.80 ± 0.01	-5.63 ± 0.05	-5.73 ± 0.07	$\mathbf{-5.99 \pm 0.07}$
Power	4.02 ± 0.04	4.11 ± 0.17	4.23 ± 0.09	$\mathbf{2.79 \pm 0.09}$	2.80 ± 0.01	2.79 ± 0.04	2.81 ± 0.07	$\mathbf{2.44 \pm 0.02}$
Protein	$\mathbf{4.36 \pm 0.01}$	4.71 ± 0.06	4.64 ± 0.03	$\mathbf{4.13 \pm 0.32}$	2.89 ± 0.00	2.83 ± 0.02	$\mathbf{2.63 \pm 0.00}$	$\mathbf{2.55 \pm 0.14}$
Yacht	1.11 ± 0.09	1.58 ± 0.48	1.57 ± 0.56	$\mathbf{0.75 \pm 0.18}$	1.55 ± 0.03	1.18 ± 0.21	1.03 ± 0.19	$\mathbf{0.62 \pm 0.11}$

The regularization coefficient λ strikes a balance between the inflation of uncertainty and model fit. If λ is set to 0, the resulting estimate may be overly confident, while setting λ too high could lead to excessive inflation. During training, the parameters m of target distribution is generated by the proposed model, with $\boldsymbol{m}_i$ being generated by the function $f(\boldsymbol{x}_i; \boldsymbol{w})$. Since each target y is associated with four parameters, our proposed model has four output neurons for each target y. To make certain that the constraints on $(\beta, \alpha, \upsilon)$ are enforced, we apply a softplus activation function (since $\alpha > 1$, with an additional $+1$ added). For other parameters, linear activation is used.

4. Experiments

4.1. Toy dataset

We first validated our ideas on small dataset and compared with baseline methods. Following [9,52], the toy dataset has inputs uniformly and randomly in the range of $[-4, 4]$. For each input x, the corresponding target y is computed as $y = x^3 + \epsilon_n$, where $\epsilon_n \sim \mathcal{N}(0, 3)$. We assessed aleatoric within ± 4 and epistemic ± 6 uncertainty estimation. We evaluated three normalizing flow methods – Radial [32], Planar [32] and RealNVP flow [48] in comparison with three baselines – PBP [52], Ensembling [9] and Dropout [10]. All the models were trained with the same parameters as $\eta = 5\mathrm{e} - 3$ for Adam optimizer learning rate, batch size of 128 and train 5000 iterations, sampling based models [9,10] employed $n = 5$ samples. As shown in Fig. 2, Within the training range $[-4, 4]$, almost all methods are able to accurately predict aleatoric uncertainty. As going beyond the training range, which of greater than 4 and less than -4 depicted in Fig. 2, the epistemic uncertainty begins to increase. The methods based on the normalizing flows well bound the uncertainty in a small range near the ground truth, and the prediction of the baseline methods on the epistemic uncertainty gradually fail.

4.2. Real world datasets

In this set of experiments, we followed the same experiment setup used by [9,10]. We evaluated FlowNet with RealNvp realization in comparison with three baseline methods — Dropout [10], Ensembles [9] and Evidential [45] from the aspects of root mean squared error (RMSE) and negative log-likelihood (NLL). The results shows summary statistics in Table 1. On each data set, the top results among proposed method are shown in bold font. From Table 1, we can conclude that whether in terms of RMSE or NLL, FlowNet exceeds the baseline methods, almost in all data sets, except concrete. At the same time, we observed in the experiment that with the addition of more intermediate normalizing flow layers, the performance is further improved, and currently only one layer was employed.

4.3. Vision tasks in complex scenes

We further evaluated the effectiveness of FlowNet on more complex vision tasks that are more close to real scenes. Image depth estimation is the task of estimating the depth value (distance relative to the camera) of each pixel given a single (monocular) RGB image. This task has a wide range of applications in many fields such as virtual reality, semantic segmentation, automatic driving, and 3D reconstruction. Due to the lack of a single image for spatial information, object occlusion, movement, and the need to process high-dimensional data at the pixel level, this problem still remains challenging.

We employed NYU Depth v2 [53] dataset as training dataset. The NYU Depth v2 dataset is a large, publicly available dataset and widely used in the computer vision community as a benchmark for evaluating the performance of depth estimation algorithms. It contains diverse indoor scenes image pairs (e.g. office, libraries, etc.), where each pair consists of an RGB image and its corresponding depth map. The depth maps were acquired using a Microsoft Kinect sensor, providing high-quality, dense depth information for each image in the dataset. FlowNet use U-Net [54] as the backbone, and in order to take a full advantage of the processing for the image dataset, we combined the Glow model [35] (with Invertible 1×1 Convolutions) to get the final output. The final layer outputs a single $H \times W$ activation map in the case of regression. Following the experiment setup with [45], the FlowNet model generates four outputs, corresponding to $(\beta, \alpha, \upsilon, \gamma)$ respectively, under restrictions. For the dropout implementation, spatial dropout uncertainty sampling [37,55] was used.

We tested the model on unseen data in the subject of accuracy and predictive epistemic uncertainty. The predicted depth and predictive entropy are shown in Fig. 3 left side, for randomly selected test images. An effective measure of epistemic uncertainty should be able to detect inaccuracies in predictions, which FlowNet effectively captures while providing clear confidence estimates. In contrast, dropout significantly underestimates uncertainty and ensembling sometimes overestimates it. Fig. 3 Middle part clearly shows the inverse trend between observed error and prediction confidence. FlowNet, while being able to accurately predict epistemic uncertainty, has a prediction accuracy comparable to start-of-the-arts.

In Fig. 3 right part, we further assess the accuracy of FlowNet on uncertainty estimates. The calibration curves are calculated as described in [56], with the ideal curve being $y = x$, Approximately 90% of the time, the target falls within a 90% certainty gap, as indicated. The results reveal that dropout method tends to overestimate confidence in low-confidence scences (0.126), while evidential (0.033) and ensembling (0.048) performs better but still falls short compared to FlowNet (calibration error: 0.015).

4.4. Resilience against adversarial examples

We then examined the scenario of OOD detection where inputs are deliberately altered to produce incorrect predictions. To generate

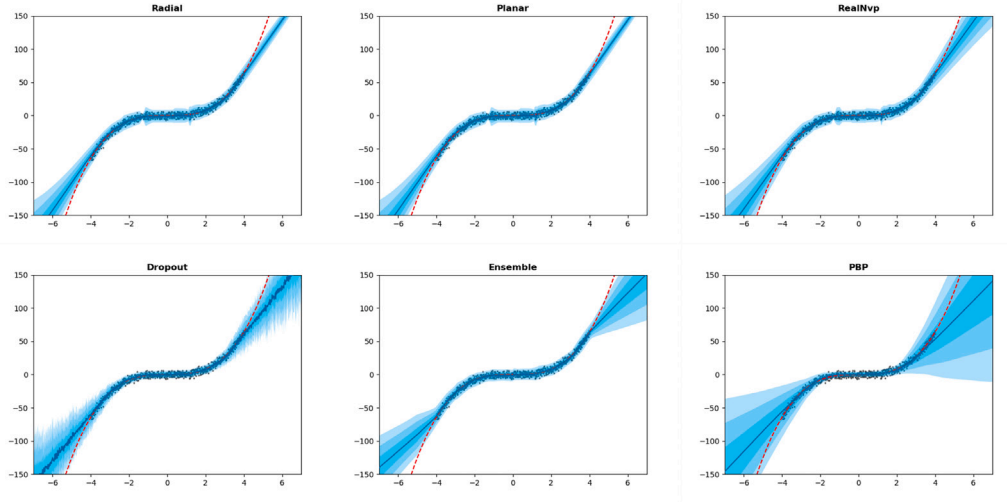


Fig. 2. Toy dataset uncertainty estimation trained on $y = x^3 + \epsilon_n, \epsilon_n \sim \mathcal{N}(0, 3)$. The top three graphs are the uncertainty estimation of FlowNet based on various normalizing flows, and the bottom three graphs are the baseline methods. FlowNet is capable of bounding the epistemic uncertainty near the ground truth, whereas baseline methods were less accurate in prediction of epistemic uncertainty.

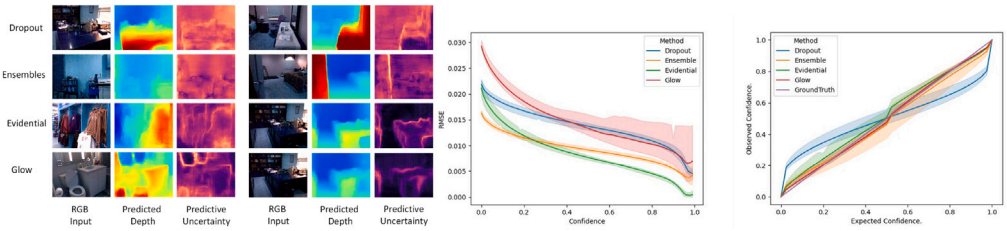


Fig. 3. Illustration of epistemic uncertainty in depth estimation. (Left) An illustration of depth predictions and the estimation of uncertainty at the pixel level. (Middle) Relationship between observed error and prediction confidence level; usually inverse trend is desired. (Right) With inset shows calibration errors, model uncertainty calibration [56], where the ideal relationship between predicted uncertainty and actual uncertainty is $y = x$.

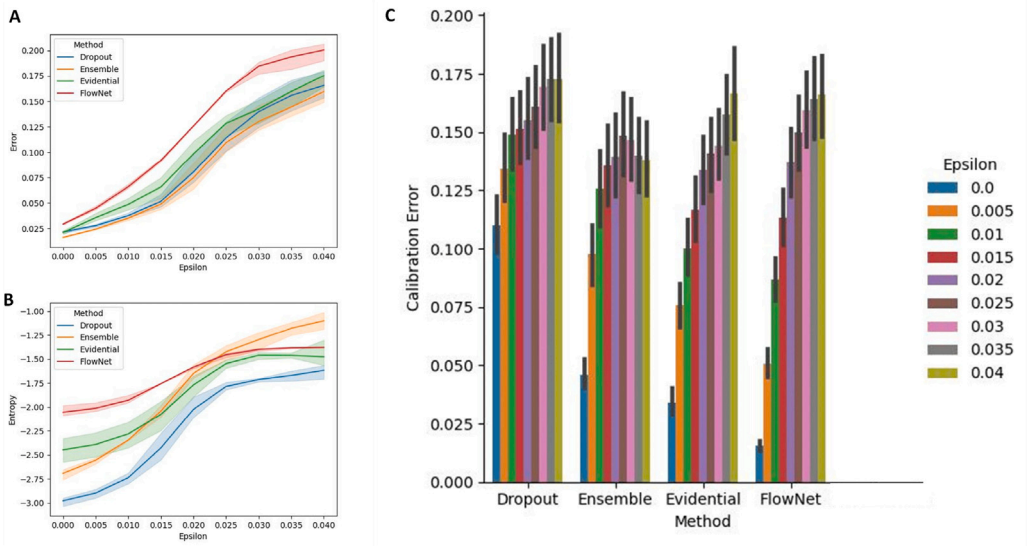


Fig. 4. The robustness of uncertainty estimates under adversarial noise is explored. The relationship between adversarial noise and both the estimated epistemic uncertainty (B) and predictive error (A) is studied. (C) The calibration performance of various methods is compared visually as the noise level increases. FlowNet exhibits the highest calibration performance among the baseline methods.

adversarial perturbations for our test set, we employed FGSM algorithm (detailed in [57]) with gradually increasing levels of noise, represented by Epsilon (ϵ). It is important to note that this experiment was not aimed at presenting a solution for advanced adversarial attacks, but rather to showcase that FlowNet accurately reflects heightened predictive uncertainty on samples that have undergone adversarial manipulations.

The results in Fig. 4 A show that as adversarial noise is added, the absolute error of all methods increases. Additionally, Fig. 4 B indicates that there is a positive effect of noise on our predictive uncertainty

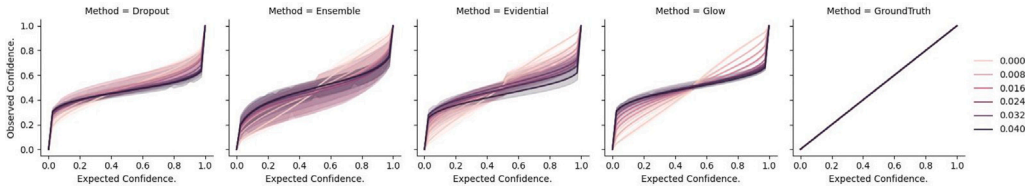


Fig. 5. The relationship between Expected Confidence and Observed Confidence of FlowNet and baseline methods.

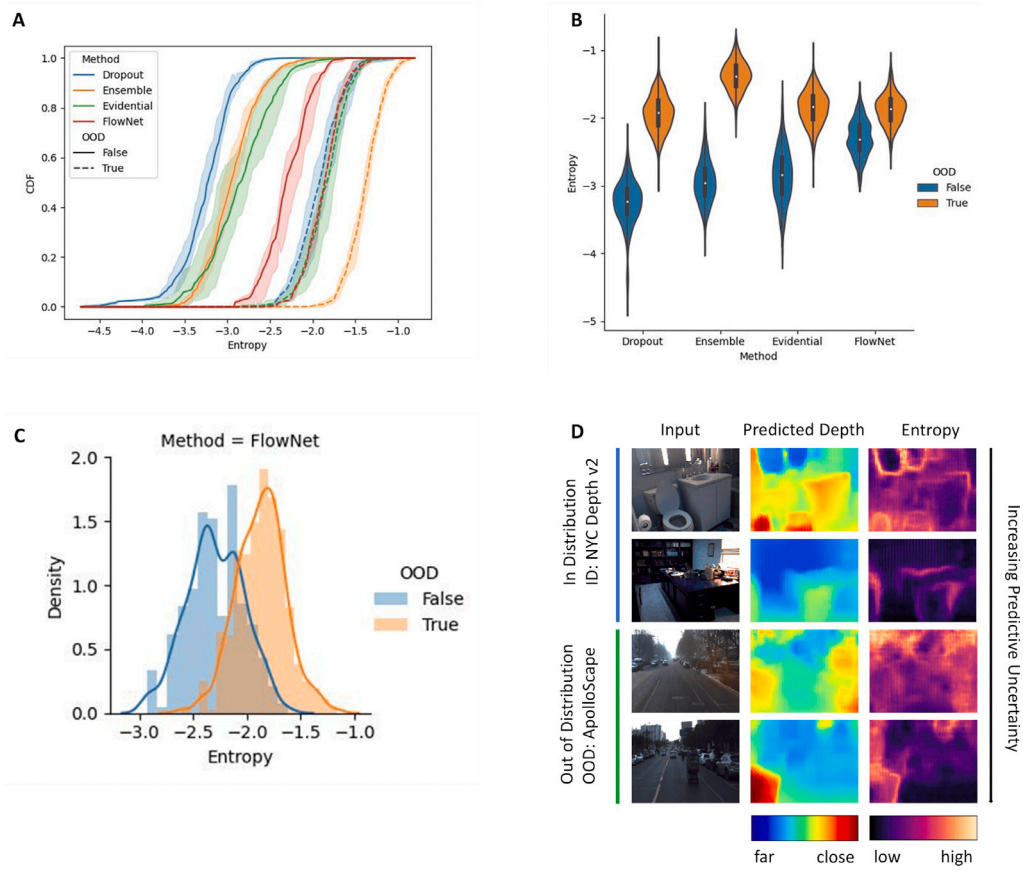


Fig. 6. The uncertainty of OOD data is analyzed. FlowNet shows low uncertainty (entropy) on ID data and amplifies uncertainty on OOD data. (**A**) presents the cumulative density function (CDF) of ID and OOD entropy for the tested methods, and OOD detection was evaluated using AUC-ROC. (**B**) compares uncertainty (entropy) across the methods. (**C**) displays full density histograms of entropy estimated by FlowNet for ID and OOD data. (**D**) Examples of predictions including both ID and OOD data.

estimates. However, as noise levels continue to rise beyond a certain threshold, the ensemble method appears to exhibit better performance in terms of predictive uncertainty. This observation underscores the ensemble method's robustness to high noise levels, possibly due to its inherent diversity among multiple models, which can provide a broader perspective on uncertainty. Fig. 4 C compares the calibration performance of various methods visually as the noise level increases. FlowNet achieves the best calibration performance compared to other baseline methods. Fig. 5 shows the relationships between expected confidence and observed confidence. The regression model based on normalizing flow controls the uncertainty in a smaller range compared with other baseline methods.

4.5. OOD sample testing

The purpose of estimating uncertainty is to determine when a ML model encounters test samples that are not part of its training distribution or when its prediction cannot be relied upon. This section looks into the capacity of FlowNet in dealing with heightened epistemic uncertainty in the case of OOD data, as evaluated on the ApolloScape [58] OOD dataset for outdoor driving scenes. It is important to emphasize that other techniques like Prior Networks [42,43] feel necessity for OOD data to further guide the identification of instances with high uncertainty during the training process, while FlowNet only replies on ID data during training and does not have this restriction.

In order to test the model, we input both ID and OOD test datasets and further documented average entropy predicted for each test image. Fig. 6 **A** displays for each test set and method, the entropy of the cumulative density function. All models performed as expected, with a positive shift, among all the models, FlowNet is competitive as shown in the results. The distribution of entropy is summarized in Fig. 6 **B** using violin plots, again highlighting the clear distinction in uncertainty on OOD data. Fig. 6 **C** shows the density distribution of the ID and OOD data and Fig. 6 **D** provides examples of predictions (both ID and OOD). These results indicate that FlowNet, without having OOD data during training, can effectively capture increased uncertainty on OOD data, matching the performance of established epistemic uncertainty estimation benchmarks.

5. Conclusion

In this paper we have presented FlowNet, a framework for uncertainty estimation in regression without requiring OOD samples for training or costly sampling for uncertainty estimation. FlowNet is composed of three main components: an encoder which outputs a position in a latent space, a normalizing flow which performs a density estimation in this latent space, and a noval loss for uncertainty-aware training. We have shown that FlowNet can accurately estimate epistemic uncertainty, handle complex vision tasks, and produce well-calibrated uncertainty estimates for OOD data. Furthermore, FlowNet is versatile and can be applied to a variety of regression tasks. It will be crucial to conduct future analyses using alternative options such as the log-normal or Normal-inverse-Wishart distribution. Further evaluate the effect of the selection of prior distribution on the estimated likelihood parameter. The effectiveness, scalability, and accuracy of FlowNet have the potential to facilitate rapid and precise uncertainty estimation necessary for the reliable deployment of neural networks in domains where safety-critical predictions are highly required.

CRediT authorship contribution statement

Baobing Zhang: Writing – original draft, Validation, Software, Methodology, Investigation, Formal analysis. **Wanxin Sui:** Writing – original draft, Validation, Methodology, Data curation. **Zhengwen Huang:** Writing – review & editing, Methodology, Conceptualization. **Maozhen Li:** Writing – review & editing, Methodology, Conceptualization. **Man Qi:** Writing – review & editing, Methodology, Data curation, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

References

[1] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, et al., An image is worth 16x16 words: Transformers for image recognition at scale, 2020, arXiv preprint arXiv:2010.11929.

[2] C. Godard, O. Mac Aodha, G.J. Brostow, Unsupervised monocular depth estimation with left-right consistency, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 270–279.

[3] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C.L. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray, et al., Training language models to follow instructions with human feedback, 2022, arXiv preprint arXiv:2203.02155.

[4] T. Brown, B. Mann, N. Ryder, M. Subbiah, J.D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al., Language models are few-shot learners, Adv. Neural Inf. Process. Syst. 33 (2020) 1877–1901.

[5] C. Blundell, J. Cornebise, K. Kavukcuoglu, D. Wierstra, Weight uncertainty in neural network, in: International Conference on Machine Learning, PMLR, 2015, pp. 1613–1622.

[6] W.J. Maddox, P. Izmailov, T. Garipov, D.P. Vetrov, A.G. Wilson, A simple baseline for bayesian uncertainty in deep learning, Adv. Neural Inf. Process. Syst. 32 (2019).

[7] K. Osawa, S. Swaroop, M.E.E. Khan, A. Jain, R. Eschenhagen, R.E. Turner, R. Yokota, Practical deep learning with Bayesian principles, Adv. Neural Inf. Process. Syst. 32 (2019).

[8] D. Eswaran, S. Günnemann, C. Faloutsos, The power of certainty: A dirichlet-multinomial model for belief propagation, in: Proceedings of the 2017 SIAM International Conference on Data Mining, SIAM, 2017, pp. 144–152.

[9] B. Lakshminarayanan, A. Pritzel, C. Blundell, Simple and scalable predictive uncertainty estimation using deep ensembles, Adv. Neural Inf. Process. Syst. 30 (2017).

[10] Y. Gal, Z. Ghahramani, Dropout as a bayesian approximation: Representing model uncertainty in deep learning, in: International Conference on Machine Learning, PMLR, 2016, pp. 1050–1059.

[11] X. Li, Y. Dai, Y. Ge, J. Liu, Y. Shan, L.-Y. Duan, Uncertainty modeling for out-of-distribution generalization, 2022, arXiv preprint arXiv:2202.03958.

[12] J. Fang, Z. Wang, W. Liu, S. Lauria, N. Zeng, C. Prieto, F. Sikström, X. Liu, A new particle swarm optimization algorithm for outlier detection: industrial data clustering in wire arc additive manufacturing, IEEE Trans. Autom. Sci. Eng. (2022).

[13] H. Li, Z. Wang, C. Lan, P. Wu, N. Zeng, A novel dynamic multiobjective optimization algorithm with non-inductive transfer learning based on multi-strategy adaptive selection, IEEE Trans. Neural Netw. Learn. Syst. (2023).

[14] D. Hendrycks, T. Dietterich, Benchmarking neural network robustness to common corruptions and perturbations, 2019, arXiv preprint arXiv:1903.12261.

[15] P.W. Koh, S. Sagawa, H. Marklund, S.M. Xie, M. Zhang, A. Balsubramani, W. Hu, M. Yasunaga, R.L. Phillips, I. Gao, et al., Wilds: A benchmark of in-the-wild distribution shifts, in: International Conference on Machine Learning, PMLR, 2021, pp. 5637–5664.

[16] Y. Gal, et al., Uncertainty in Deep Learning (Ph.D. thesis), University of Cambridge, 2016.

[17] A. Kendall, Y. Gal, What uncertainties do we need in Bayesian deep learning for computer vision? Adv. Neural Inf. Process. Syst. 30 (2017).

[18] M. Hong, J. Liu, C. Li, Y. Qu, Uncertainty-driven dehazing network, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 36, No. 1, 2022, pp. 906–913.

[19] J. Hornauer, V. Belagiannis, Gradient-based uncertainty for monocular depth estimation, in: Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XX, Springer, 2022, pp. 613–630.

[20] N. Zeng, X. Li, P. Wu, H. Li, X. Luo, A novel tensor decomposition-based efficient detector for low-altitude aerial objects with knowledge distillation scheme, IEEE/CAA J. Autom. Sin. 11 (2) (2024) 487–501.

[21] P. Wu, Z. Wang, H. Li, N. Zeng, KD-PAR: A knowledge distillation-based pedestrian attribute recognition model with multi-label mixed feature learning network, Expert Syst. Appl. 237 (2024) 121305.

[22] C. Guo, G. Pleiss, Y. Sun, K.Q. Weinberger, On calibration of modern neural networks, in: International Conference on Machine Learning, PMLR, 2017, pp. 1321–1330.

[23] J. Quinonero-Candela, M. Sugiyama, A. Schwaighofer, N.D. Lawrence, Dataset Shift in Machine Learning, Mit Press, 2008.

[24] S.G. Finlayson, A. Subbaswamy, K. Singh, J. Bowers, A. Kupke, J. Zittrain, I.S. Kohane, S. Saria, The clinician and dataset shift in artificial intelligence, N. Engl. J. Med. 385 (3) (2021) 283–286.

[25] H. Guo, H. Wang, Q. Ji, Uncertainty-guided probabilistic transformer for complex action recognition, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 20052–20061.

[26] D. Hendrycks, S. Basart, N. Mu, S. Kadavath, F. Wang, E. Dorundo, R. Desai, T. Zhu, S. Parajuli, M. Guo, et al., The many faces of robustness: A critical analysis of out-of-distribution generalization, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 8340–8349.

[27] B. Charpentier, D. Zügner, S. Günnemann, Posterior network: Uncertainty estimation without ood samples via density-based pseudo-counts, Adv. Neural Inf. Process. Syst. 33 (2020) 1356–1367.

[28] Y. Ovadia, E. Fertig, J. Ren, Z. Nado, D. Sculley, S. Nowozin, J. Dillon, B. Lakshminarayanan, J. Snoek, Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift, Adv. Neural Inf. Process. Syst. 32 (2019).

[29] I. Kononenko, Bayesian neural networks, Biol. Cybernet. 61 (5) (1989) 361–370.

[30] O. Sagi, L. Rokach, Ensemble learning: A survey, Wiley Interdiscip. Rev. Data Min. Knowl. Discov. 8 (4) (2018) e1249.

[31] G.E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, R.R. Salakhutdinov, Improving neural networks by preventing co-adaptation of feature detectors, 2012, arXiv preprint arXiv:1207.0580.

[32] D. Rezende, S. Mohamed, Variational inference with normalizing flows, in: International Conference on Machine Learning, PMLR, 2015, pp. 1530–1538.

[33] I. Kobyzev, S.J. Prince, M.A. Brubaker, Normalizing flows: An introduction and review of current methods, IEEE Trans. Pattern Anal. Mach. Intell. 43 (11) (2020) 3964–3979.

[34] J.P. Agnelli, M. Cadeiras, E.G. Tabak, C.V. Turner, E. Vanden-Eijnden, Clustering and classification through normalizing flows in feature space, Multiscale Model. Simul. 8 (5) (2010) 1784–1802.

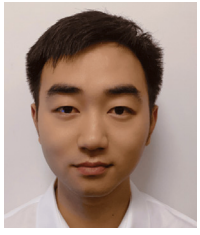
[35] D.P. Kingma, P. Dhariwal, Glow: Generative flow with invertible 1x1 convolutions, Adv. Neural Inf. Process. Syst. 31 (2018).

[36] L. Dinh, D. Krueger, Y. Bengio, Nice: Non-linear independent components estimation, 2014, arXiv preprint arXiv:1410.8516.

[37] A. Amini, A. Soleimany, S. Karaman, D. Rus, Spatial uncertainty sampling for end-to-end control, 2018, arXiv preprint arXiv:1805.04829.

[38] Y. Gal, J. Hron, A. Kendall, Concrete dropout, Adv. Neural Inf. Process. Syst. 30 (2017).

[39] D. Molchanov, A. Ashukha, D. Vetrov, Variational dropout sparsifies deep neural networks, in: International Conference on Machine Learning, PMLR, 2017, pp. 2498–2507.
[40] T. Pearce, M. Zaki, A. Brintrup, N. Anastassacos, A. Neely, Uncertainty in neural networks: Bayesian ensembling, stat 1050 (2018) 12.
[41] M. Biloš, B. Charpentier, S. Günnemann, Uncertainty on asynchronous time event prediction, Adv. Neural Inf. Process. Syst. 32 (2019).
[42] A. Malinin, M. Gales, Predictive uncertainty estimation via prior networks, Adv. Neural Inf. Process. Syst. 31 (2018).
[43] M. Gales, A. Malinin, Reverse KL-divergence training of prior networks: Improved uncertainty and adversarial robustness, 2019.
[44] M. Sensoy, L. Kaplan, M. Kandemir, Evidential deep learning to quantify classification uncertainty, Adv. Neural Inf. Process. Syst. 31 (2018).
[45] A. Amini, W. Schwarting, A. Soleimany, D. Rus, Deep evidential regression, Adv. Neural Inf. Process. Syst. 33 (2020) 14927–14937.
[46] G. Parisi, R. Shankar, Statistical Field Theory, Westview Press, 1988.
[47] M. Jordan, The exponential family: Conjugate priors, 2009.
[48] L. Dinh, J. Sohl-Dickstein, S. Bengio, Density estimation using real NVP, 2016, arXiv preprint arXiv:1605.08803.
[49] G. Papamakarios, T. Pavlakou, I. Murray, Masked autoregressive flow for density estimation, Adv. Neural Inf. Process. Syst. 30 (2017).
[50] C.-W. Huang, D. Krueger, A. Lacoste, A. Courville, Neural autoregressive flows, in: International Conference on Machine Learning, PMLR, 2018, pp. 2078–2087.
[51] D.P. Kingma, T. Salimans, R. Jozefowicz, X. Chen, I. Sutskever, M. Welling, Improved variational inference with inverse autoregressive flow, Adv. Neural Inf. Process. Syst. 29 (2016).
[52] J.M. Hernández-Lobato, R. Adams, Probabilistic backpropagation for scalable learning of bayesian neural networks, in: International Conference on Machine Learning, PMLR, 2015, pp. 1861–1869.
[53] N. Silberman, D. Hoiem, P. Kohli, R. Fergus, Indoor segmentation and support inference from RGBD images, ECCV (5) 7576 (2012) 746–760.
[54] O. Ronneberger, P. Fischer, T. Brox, U-net: Convolutional networks for biomedical image segmentation, in: Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18, Springer, 2015, pp. 234–241.
[55] J. Tompson, R. Goroshin, A. Jain, Y. LeCun, C. Bregler, Efficient object localization using convolutional networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 648–656.
[56] V. Kuleshov, N. Fenner, S. Ermon, Accurate uncertainties for deep learning using calibrated regression, in: International Conference on Machine Learning, PMLR, 2018, pp. 2796–2804.
[57] I.J. Goodfellow, J. Shlens, C. Szegedy, Explaining and harnessing adversarial examples, 2014, arXiv preprint arXiv:1412.6572.
[58] X. Huang, X. Cheng, Q. Geng, B. Cao, D. Zhou, P. Wang, Y. Lin, R. Yang, The apolloscape dataset for autonomous driving, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2018, pp. 954–960.

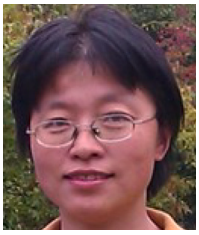


Baobing Zhang received his Ph.D. degree in Artificial Intelligence from Brunel University London, UK in 2020. He is currently a Post-Doctoral Research Fellow at Brunel University London. His research interests include deep learning, computer vision, image processing, data privacy, and AI applications.

Wanxin Sui is currently a Ph.D. candidate at Brunel University London, UK. Her research interests are in the areas of data protection and privacy, privacy preserving Al techniques and Al applications in higher education.

Zhengwen Huang received the M.Sc. degree from King's College London, London, U.K., and the Ph.D. degree from the Department of Electronic and Electrical Engineering, Brunel University London, Uxbridge, U.K. in 2014. He is currently a full member of Brunel Interdisciplinary Power Systems and a senior Research Fellow with Systems Engineering Research Group, Brunel University. He is the Leader of Artificial Intelligence and System Optimization Group, BUL-CQUPT Innovation Centre, Chongqing University of Posts and Telecommunications, Chongqing, China. His research interests include evolutionary algorithms (gene expression programming, genetic programming) and data engineering.

Maozhen Li is a Professor in the Department of Electronic and Electrical Engineering, Brunel University London, UK. He received the Ph.D. from the Institute of Software, Chinese Academy of Sciences in 1997. His main research interests are in the areas of high performance computing, big data analytics and artificial intelligence with applications to smart grid, smart manufacturing and smart cities. He has over 200 research publications in these areas including 4 books and 120 peer reviewed journal papers. He has served over 30 IEEE conferences and is on the editorial board of a number of journals. He is a Fellow of the British Computer Society (BCS) and the Institute of Engineering and Technology (IET).



Man Qi is a Reader in Computing at Canterbury Christ Church University. Her research interests are in Cyber Security, Data Intelligence, IoT and HCI. Dr Qi published over 80 research papers including over 30 journal papers and is the editorial board member for 5 international journals. She has been the PhD external examiners for many Universities in Australia and UK. Dr Qi has served as chair/program committee member for around 50 international conferences and been long term reviewer for many international journals. She is Fellow of British Computer Society (FBCS) and Fellow of Higher Education Academy (FHEA).