DEEP EVIDENTIAL REGRESSION

Anonymous authors
Paper under double-blind review

ABSTRACT

Deterministic neural networks (NNs) are increasingly being deployed in safety critical domains, where calibrated, robust and efficient measures of uncertainty are crucial. While it is possible to train regression networks to output the parameters of a probability distribution by maximizing a Gaussian likelihood function, the resulting model remains oblivious to the underlying confidence of its predictions. In this paper, we propose a novel method for training deterministic NNs to not only estimate the desired target but also the associated evidence in support of that target. We accomplish this by placing evidential priors over our original Gaussian likelihood function and training our NN to infer the hyperparameters of our evidential distribution. We impose priors during training such that the model is penalized when its predicted evidence is not aligned with the correct output. Thus the model estimates not only the probabilistic mean and variance of our target but also the underlying uncertainty associated with each of those parameters. We observe that our evidential regression method learns well-calibrated measures of uncertainty on various benchmarks, scales to complex computer vision tasks, and is robust to adversarial input perturbations.

1 INTRODUCTION

Recent advances in deep supervised learning have yielded super human level performance and precision (Liu et al., 2015; Gebru et al., 2017). While these models empirically generalize well when placed into new test enviornments, they are often easily fooled by adversarial perturbations (Goodfellow et al., 2014), and have difficulty understanding when their predictions should not be trusted. Today, regression based neural networks are being deployed in safety critical domains of computer vision (Godard et al., 2017; Alahi et al., 2016) as well as in robotics and control (Bojarski et al., 2016) where the ability to infer model uncertainty is crucial for eventual wide-scale adoption. Furthermore, precise uncertainty estimates are useful both for human interpretation of confidence and anomaly detection, and also for propagating these estimates to other autonomous components of a larger, connected system.

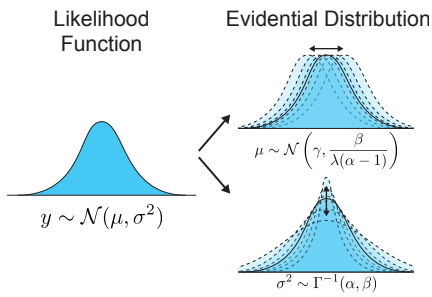

Figure 1: **Evidential distributions.** Maximum likelihood optimization learns a likelihood distribution given data, while evidential distributions model higher-order probability distribution over the likelihood parameters.

Existing approaches to uncertainty estimation are roughly split into two main categories: (1) learning aleatoric uncertainty (uncertainty in the data) and (2) epistemic uncertainty (uncertainty in the prediction). While representations for aleatoric uncertainty can be learned directly from data, approaches for estimating epistemic uncertainty primarily focus on placing probabilistic priors over all weights and sampling many times to obtain a measure of variance. In practice, many challenges arise with this approach, such as the computational expense of sampling during inference, how to pick an appropriate weight prior, or even how to learn such a representation given your prior.

We approach the problem of uncertainty estimation in regression from an evidential state of mind, where the model can acquire evidence during learning as it sees training examples. Every training example adds support to a learned higher-order, *evidential* distribution. Sampling from this evidential distribution yields instances of lower-order, likelihood functions from which the data was drawn (cf. Fig. 1). We demonstrate that, by placing priors over our likelihood function (instead of all weights),

we can learn a grounded representation of epistemic and aleatoric uncertainty that can be computed without sampling during inference.

In summary, this work makes the following contributions:

1. A novel and scalable method for learning representations of epistemic and aleatoric uncertainty, specifically on regression problems, by placing evidential priors over our likelihood function;
2. Evaluation of learned epistemic uncertainty on benchmark regression tasks and comparison against other state-of-the-art uncertainty estimation techniques for neural networks;
3. Robustness evaluation against out of distribution and adversarially perturbed test data.

2 MODELLING UNCERTAINTIES FROM DATA

2.1 PRELIMINARIES

Consider the following supervised optimization problem: given a dataset, $\mathcal{D}$, of N paired training examples, $(x_1, y_1), \ldots, (x_N, y_N)$, we aim to learn a function f, parameterized by a set of weights, $\boldsymbol{w}$, which approximately solves the following optimization problem:

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

where $L_i(\cdot)$ describes a loss function. In this work, we consider deterministic regression problems, which commonly optimize the sum of squared errors, $L_i(\boldsymbol{w}) = \frac{1}{2} \|y_i - f(x_i; \boldsymbol{w})\|^2$. In doing so, the model is encouraged to learn the average correct answer for a given input, but does not explicitly model any underlying noise or uncertainty in the data when making its estimation.

2.2 MAXIMUM LIKELIHOOD ESTIMATION

Alternatively, we can approach our optimization problem from a maximum likelihood perspective, where we learn model parameters that maximize the likelihood of observing the particular set of training datapoints. In the context of deterministic regression, if we assume our targets, y_i, were drawn i.i.d. from a Gaussian distribution with mean and variance parameters $\theta = (\mu, \sigma^2)$, then the likelihood of observing a single target, y_i, can be expressed as

$$p(y_i|\theta) = p(y_i|\mu, \sigma^2) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left\{-\frac{1}{2\sigma^2}(y_i - \mu)^2\right\}. \tag{2}$$

In maximum likelihood estimation, we aim to learn a model to infer the (μ, σ^2) that maximize the likelihood of observing our targets, y. Equivalently, instead of maximizing the likelihood function, in practice it is common to instead minimize the negative log likelihood by setting

$$L_i(\boldsymbol{w}) = -\log\, p(y_i|\mu, \sigma^2) = \frac{1}{2}\log(2\pi\sigma^2) + \frac{(y_i - \mu)^2}{2\sigma^2}. \tag{3}$$

In learning the parameters θ, this likelihood function allows us to successfully model the uncertainty of our data, also known as the aleatoric uncertainty. However, our model remains oblivious to the predictive model uncertainty. This metric, known as epistemic uncertainty, corresponds to the model's uncertainty in its own output prediction (Kendall & Gal, 2017). In this paper, we present a novel approach for estimating the evidence in support of network predictions by directly learning both the inferred aleatoric uncertainty as well as the underlying epistemic uncertainty over its predictions. We achieve this by placing higher-order prior distributions over the learned parameters governing the distribution from which our observations are drawn.

3 EVIDENTIAL UNCERTAINTY FOR REGRESSION

3.1 PROBLEM SETUP

We consider the problem where our observed targets, y_i, are drawn i.i.d. from a Gaussian distribution with *unknown mean and variance* (μ, σ^2), which we seek to probabilistically estimate. We model

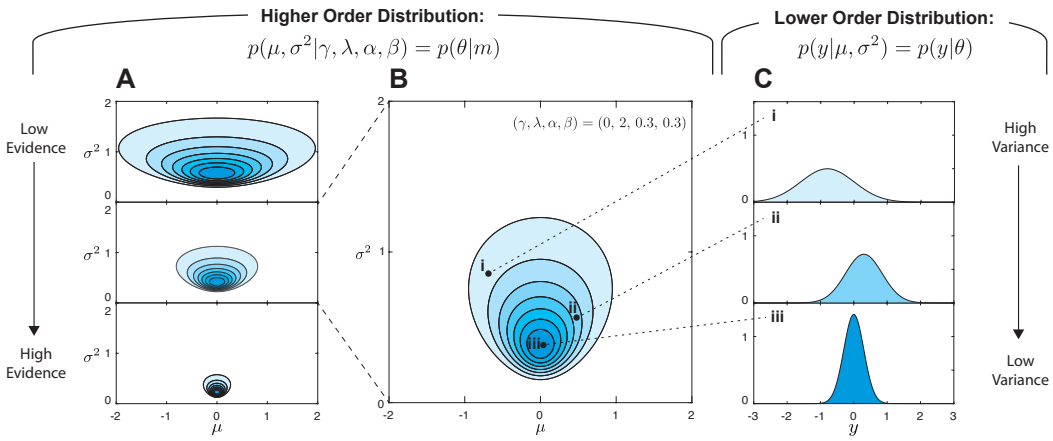

Figure 2: **Normal Inverse-Gamma distribution.** Different realizations of our evidential distribution (A) correspond to different levels of confidences in the parameters (e.g. μ, σ^2). Sampling from a single realization of a higher-order evidential distribution (B), yields lower-order distributions (C) over the data (e.g. $p(y|\mu, \sigma^2)$). Darker shading indicates higher probability mass.

this by placing a conjugate prior distribution on (μ, σ^2). If we assume our observations are drawn from a Gaussian, this leads to placing a Gaussian prior on our unknown mean and an Inverse-Gamma prior on our unknown variance:

$$\begin{aligned}(y_1, \ldots, y_N) &\sim \mathcal{N}(\mu, \sigma^2) \\ \mu &\sim \mathcal{N}(\gamma, \sigma^2\lambda^{-1}) \\ \sigma^2 &\sim \Gamma^{-1}(\alpha, \beta).\end{aligned} \tag{4}$$

where $\gamma \in R$, $\lambda > 0$, $\alpha > 0$, $\beta > 0$.

From a variational Bayesian perspective, our aim is to estimate a posterior distribution $q(\mu, \sigma^2) = p(\mu, \sigma^2|y_1, \ldots, y_N)$ of the parameters (μ, σ^2). To obtain an approximation for the true posterior, we assume that the estimated distribution can be factorized into independent factors such that $q(\mu, \sigma^2) = q(\mu)\, q(\sigma^2)$. In this case, the true distribution takes the form of a Normal Inverse-Gamma (N.I.G.) distribution:

$$p(\mu, \sigma^2|\gamma, \lambda, \alpha, \beta) = \frac{\beta^\alpha \sqrt{\lambda}}{\Gamma(\alpha)\sqrt{2\pi\sigma^2}} \left(\frac{1}{\sigma^2}\right)^{\alpha+1} \exp\left(-\frac{2\beta + \lambda(\gamma - \mu)^2}{2\sigma^2}\right). \tag{5}$$

The mean of this distribution can be interpreted as being estimated from λ observations with sample mean γ while its variance was estimated from 2α observations with sample mean γ and sum of squared deviations 2β. We denote the *total evidence* as the sum of all inferred observations counts, $\sum_j \phi_j = \lambda + 2\alpha$.

Thus, we can interpret the estimated posterior $q(\theta) = q(\mu, \sigma^2)$ as an *evidential*, *higher-order* probability distribution on top of the unknown *lower-order* likelihood distribution from which observations are drawn. Drawing a single sample θ_j from our evidential posterior yields (μ_j, σ_j^2) representing a single instance of our likelihood function, namely $\mathcal{N}(\mu_j, \sigma_j^2)$. Thus, the parameters of the posterior, specifically $(\gamma, \lambda, \alpha, \beta)$, determine not only the location but also the dispersion concentrations, or uncertainty, associated with our inferred likelihood function.

For example, in Fig. 2A we visualize the posterior of different evidential N.I.G. distributions with varying model parameters. We illustrate that by increasing the evidential parameters (i.e. λ, α) of this distribution, the p.d.f. becomes more tightly concentrated about its inferred likelihood function. Considering a single parameter realization of this higher-order distribution, cf. Fig. 2B, we can subsequently sample many lower-order realizations of our likelihood function, as shown in Fig. 2C.

In this work, we use neural networks to learn a higher-order evidential distribution that directly captures prediction uncertainties and evaluate our method on various regression tasks. This approach presents several distinct advantages. First, this method enables simultaneous learning of the desired regression task, along with uncertainty estimation, built in, due to our evidential priors. Second,

since the evidential prior is a higher-order N.I.G. distribution, the maximum likelihood Gaussian can be computed analytically from the expected values of the (μ, σ^2) parameters, without the need for sampling. Third, by explicitly modeling the evidence, we effectively capture the epistemic or model uncertainty associated with the network's prediction. This can be done by simply evaluating the variance of our inferred evidential distribution.

3.2 Learning the evidential distribution

Having formalized the problem of using an evidential distribution to capture model uncertainty, we next describe our approach for actually learning this distribution. Given a set of observations, variational inference methods aim to approximate a posterior distribution over unobserved variables or parameters by maximizing the evidence lower bound (ELBO) (Kingma & Welling, 2013). Similarly, here we seek to estimate the posterior distribution $q(\theta) = q(\mu, \sigma^2)$ governed by the higher-order distribution parameters $m = (\gamma, \lambda, \alpha, \beta)$ to maximize the likelihood of our observations. Applying the principle of variational inference, we have:

$$\text{ELBO} := \underbrace{\mathbb{E}_q[\log p_m(y|\theta)]}_{\text{log-likelihood}} - \underbrace{\text{KL}[\, q(\theta|y) \,\|\, p(\theta)\,]}_{\text{dissimilarity penalty}} \tag{6}$$

where $\theta = (\mu, \sigma^2)$. Similar to the principle of variational inference, in the remainder of this section we will discuss how we learn evidential distributions for regression by maximizing the log-likelihood of model evidence and minimizing the distance to an uncertainty prior. As we will see, maximizing the log-likelihood allows our model to fit the data, while the regularization provides an "uncertainty" penalty so the model can express when it does not know the answer.

We define the "model evidence" as the likelihood of the observations, y, given the evidential distribution parameters m, $p(y|m)$. We apply Bayes' theorem and marginalize over the likelihood parameters θ to obtain an equation for the model evidence:

$$p(y|m) = \frac{p(y|\theta, m)p(\theta|m)}{p(\theta|y, m)} = \int_\theta p(y|\theta, m)p(\theta|m)\, d\theta. \tag{7}$$

The model evidence is not, in general, straightforward to evaluate since computing it involves integrating out the dependence on model parameters. However, by placing a N.I.G. prior on our Gaussian likelihood function an analytical solution does exist.

$$p(y|m) = \int_{\sigma^2=0}^{\infty} \int_{\mu=-\infty}^{\infty} \left[\sqrt{\frac{1}{2\pi\sigma^2}} e^{-\frac{1}{2\sigma^2}(y-\mu)^2}\right] \left[\frac{\beta^\alpha \sqrt{\lambda}}{\Gamma(\alpha)\sqrt{2\pi}} \tau^{\alpha-\frac{1}{2}} e^{-\frac{\beta}{\sigma^2}} e^{-\frac{\lambda\pi(\mu-\gamma)^2}{2}}\right] \mathrm{d}\mu\, \mathrm{d}\sigma^2 \tag{8}$$

For computational reasons it is common to instead minimize the negative log-likelihood of the model evidence (NLL_{ME}). For a complete derivation please refer to the appendix.

$$\text{NLL}_{\text{ML}} = -\log p(y|m) = -\log\left(2^{\frac{1}{2}+\alpha}\beta^\alpha \sqrt{\frac{\lambda}{2\pi(1+\lambda)}} \left(2\beta + \frac{\lambda(\gamma-y)^2}{1+\lambda}\right)^{-\frac{1}{2}-\alpha}\right) \tag{9}$$

Alternatively, we can also derive the negative log likelihood of model evidence from the sum-of-square deviations to compute NLL_{SOS}:

$$p(y|m) = \int_{\sigma^2=0}^{\infty} \int_{\mu=-\infty}^{\infty} \mathbb{E}_{\hat{y}\sim p(y|\mu,\sigma^2)}\left[||\hat{y}-y||_2^2\right] p(\mu, \sigma^2|\gamma, \lambda, \alpha, \beta)\ \mathrm{d}\mu\, \mathrm{d}\sigma^2 \tag{10}$$

$$\text{NLL}_{\text{SOS}} = \log\left(\frac{\beta(1+\lambda)}{\lambda} + (\alpha-1)(y-\gamma)^2\right) + \log\left(\frac{\Gamma(\alpha-1)}{\Gamma(\alpha)}\right) \tag{11}$$

In our experiments, using NLL_{SOS} resulted in greater training stability and increased performance, as opposed to the NLL_{ML} loss. Therefore, the remainder of this paper present results using NLL_{SOS}.

3.3 Expressing "I don't know"

In the previous subsection, we outlined a loss function for training a NN to output parameters of a N.I.G. distribution which maximize the log-likelihood of our data. In this subsection we describe how

we regularize training against a prior where the model does not have any evidence (i.e. maximum uncertainty). In variational inference this is done by minimizing the KL-divergence between the inferred posterior, $q(\mu, \sigma^2) = q(\theta)$, and a prior, $p(\theta)$, cf. Eq. 6. In the evidential setting, our prior is also a Normal Inverse-Gamma distribution, but with zero evidence (or infinite uncertainty). Therefore, during training we aim to minimize our evidence (or maximize our uncertainty) everywhere except where we have training data, as enforced by the negative log-likelihood loss term.

Unfortunately, the KL-divergence between an arbitrary N.I.G. distribution and another with infinitely low evidence is not well defined (Soch & Allefeld, 2016). To address this, we formulate a custom evidence regularizer, R_i, based on the error of the i-th prediction,

$$R = \|y - \gamma\|_p \cdot \sum_j \phi_j = \|y - \gamma\|_p \cdot (2\alpha + \lambda),$$

where $\|x\|_p$ represents the L-p norm of x.

This regularization loss imposes a penalty whenever there is an error in the prediction that scales with the total evidence of our inferred posterior. Conversely, large amounts of predicted evidence will not be penalized as long as the prediction is close to the target observation.

The final combined loss function for training consisting of the negative log-likelihood of model evidence and and the regularization loss is

$$\mathcal{L} = \text{NLL}_{\text{SOS}} + R. \tag{12}$$

3.4 Evaluating aleatoric and epistemic uncertainty

The aleatoric uncertainty, also referred to as statistical or data uncertainty, is representative of unknowns that differ each time we run the same experiment. We evaluate the aleatoric uncertainty from $\mathbb{E}[\sigma^2] = \frac{\beta}{\alpha - 1}$. The epistemic, also known as the model uncertainty, describes the estimated uncertainty in the learned model and is defined as $\text{Var}[\mu] = \frac{\beta}{(\alpha-1)\lambda}$, based on the N.I.G. definition.

4 Experiments

4.1 Predictive accuracy and uncertainty benchmarking

We first qualitatively compare the performance of our approach against a set of benchmarks on a one-dimensional toy regression dataset. The training set consists of training examples drawn from $y = \sin(3x)/(3x) + \epsilon$, where $\epsilon \sim \mathcal{N}(0, 0.02)$ in the region $-3 \leq x \leq 3$, whereas the test data is unbounded. Not only determinisitic or maximum likelihood regression, but also techniques using empirical variance of the networks' predictions such as MC-dropout, model-ensembles, and Bayes-by-Backprop underestimate the uncertainty outside the training distribution. In contrast, evidential regression estimates uncertainty appropriately and grows the uncertainty estimate with increasing distance from the training data (Figure 3).

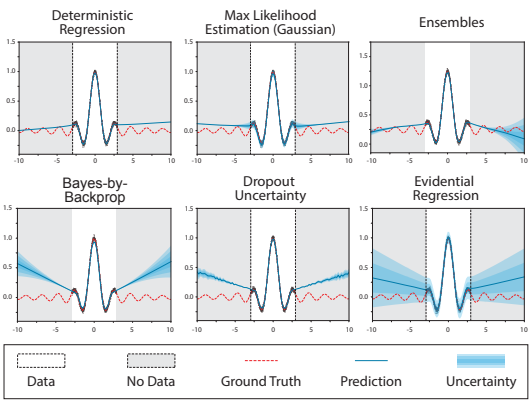

Figure 3: **Uncertainty estimation in regression.** Capturing the uncertainty of neural networks is a core challenge of regression learning, especially when presented out-of-distribution data. Modeling the supportive evidence during learning enables precise estimation within the training regime and conservative uncertainty estimates where there was no training data.

Additionally, we compare our approach to state-of-the-art methods for predictive uncertainty estimation using NNs on common real world datasets used in (Hernández-Lobato & Adams, 2015; Lakshminarayanan et al., 2017; Gal & Ghahramani, 2016). We evaluate our proposed evidential regression method against model-ensembles and BBB based on root mean squared error (RMSE), and negative log-likelihood (NLL). We do not provide results for MC-dropout since it consistently performed inferior to the other baselines. The results in Table 1 indicate that although the loss function for evidential regression is more complex than competing approaches, it is the top performer in RMSE and NLL in 8 out of 9 datasets.

Dataset	RMSE Ensembles	RMSE BBB	RMSE Evidential	NLL Ensembles	NLL BBB	NLL Evidential
Boston	0.09 ± 4.3e-4	0.09 ± 3.7e-4	**0.09 ± 1.0e-6**	**-0.89 ± 6.5e-2**	-0.67 ± 1.5e-2	**-0.87 ± 2.2e-2**
Concrete	**0.07 ± 4.4e-3**	**0.06 ± 3.3e-6**	**0.06 ± 7.0e-7**	**-1.29 ± 4.1e-2**	**-1.32 ± 4.3e-3**	**-1.31 ± 1.9e-2**
Energy	**0.10 ± 2.3e-4**	**0.10 ± 1.6e-5**	**0.10 ± 9.0e-7**	-0.61 ± 8.9e-2	-0.60 ± 2.0e-2	**-0.75 ± 1.4e-2**
Kin8nm	**0.07 ± 3.5e-4**	0.17 ± 3.5e-4	0.08 ± 3.8e-3	-0.78 ± 1.4e-2	-0.32 ± 6.3e-3	**-1.17 ± 2.6e-2**
Naval	**0.01 ± 1.0e-7**	0.04 ± 1.2e-2	**0.01 ± 3.4e-4**	-2.55 ± 3.3e-2	-1.83 ± 2.4e-1	**-3.17 ± 2.1e-3**
Power	**0.06 ± 4.0e-7**	**0.06 ± 2.3e-6**	**0.06 ± 5.3e-6**	-1.29 ± 6.9e-2	-1.33 ± 2.5e-3	**-1.40 ± 6.2e-3**
Protein	**0.17 ± 1.0e-6**	0.17 ± 8.0e-4	**0.17 ± 1.6e-6**	**-0.27 ± 6.7e-2**	0.32 ± 5.9e-2	**-0.29 ± 1.1e-2**
Wine	**0.10 ± 3.0e-4**	**0.10 ± 2.9e-4**	**0.10 ± 3.8e-5**	-0.46 ± 2.5e-1	**-0.89 ± 2.4e-3**	-0.85 ± 6.9e-3
Yacht	0.07 ± 1.3e-3	0.07 ± 3.4e-3	**0.06 ± 6.2e-5**	-1.16 ± 6.3e-2	-0.74 ± 5.8e-2	**-1.28 ± 9.4e-3**

Table 1: **Test performance for benchmark regression tasks.** We evaluate RMSE and predictive negative log-likelihood (NLL) for model ensembling (Lakshminarayanan et al., 2017), Bayes-By-Backprop (BBB) (Graves, 2011), and our method, evidential regression. Evidential modeling achieves top statistics for both RMSE and NLL in eight of the nine datasets tested. Top scores (within statistical significance) are bolded in the table.

4.2 Depth estimation

After establishing benchmark comparison results, in this subsection we demonstrate the scalability of our evidential learning by extending to the complex, high-dimensional task of depth estimation. Monocular end-to-end depth estimation is a central problem in computer vision which aims to learn a representation of depth directly from an RGB image of the scene. This is a challenging learning task since the output target y is very high-dimensional, $y \in \mathbb{R}^{H \cdot W}$, where (H, W) are the height and width of the input image respectively. For every pixel in the image we regress over the desired depth and simultaneously want to estimate the uncertainty associated to that individual pixel estimate.

Our training data consists of over 27k RGB-to-depth pairs of indoor scenes (e.g. kitchen, bedroom, etc) from the NYU Depth v2 dataset (Nathan Silberman & Fergus, 2012). We train a U-Net style NN (Ronneberger et al., 2015) for inference. Spatial dropout (Tompson et al., 2015) (with $p = 0.1$) is used for the dropout baseline. The final layer of our model outputs a single $H \times W$ activation map in the case of deterministic regression, dropout, ensembling and BBB. However, for our evidential model, we infer four $H \times W$ outputs, each corresponding to $(\gamma, \lambda, \alpha, \beta)$ respectively.

Table 2 summarizes the size and speed of all models. Evidential models contain significantly fewer trainable parameters than ensembles (where the number of parameters scales linearly with the size of the ensemble). BBB maintains a trainable mean and variance for every weight in the network, so its size is roughly $2\times$ larger as well. The number of trainable parameters for evidential regression is closest to that of dropout, which has fewer as it contains a slightly smaller final output layer. Since evidential regression models do not require sampling in order to estimate their uncertainty, their forward-pass inference times are also significantly more efficient when compared to the baselines. Finally, we demonstrate that we achieve comparable predictive accuracy (through RMSE and NLL) to the other models. Note that the output size of the depth estimation problem presented significant learning challenges for the BBB baseline, and it was unable to converge during training. As a result, for the remainder of this analysis we compare against only spatial dropout and ensembles.

We evaluate these models in terms of both their accuracy and their predictive uncertainty on previously unseen test set examples. Fig. 4A-C visualizes the predicted depth, absolute error from ground truth, and predictive uncertainty across three randomly picked test images. Ideally, a strong predictive uncertainty would capture any errors in the prediction (i.e., roughly correspond to where the model is making errors). We note that, compared to dropout and ensembling approaches, evidential uncertainty modeling captures the depth errors while providing clear and localized predictions of confidence,

	Parameters		**Inference Speed**		**RMSE**	**NLL**
	Absolute	Relative	Seconds	Relative		
Evidential (Ours)	7,846,776	1	0.013	1	0.02 ± 0.04	-1.05 ± 0.35
Spatial Dropout	7,846,657	0.99	0.093	7.21	0.03 ± 0.03	-1.22 ± 0.46
Ensembles	39,233,285	4.99	0.071	5.49	0.03 ± 0.03	-0.99 ± 0.28
BBB	15,693,314	1.99	0.102	7.84	-	-

Table 2: **Depth estimation performance.** Comparison of different epistemic uncertainty estimation algorithms and predictive performance on an unseen test set. For fair comparison, dropout, ensembles, and Bayes-by-Backprop were all sampled 5 times, which provided the best speed-space-accuracy tradeoff.

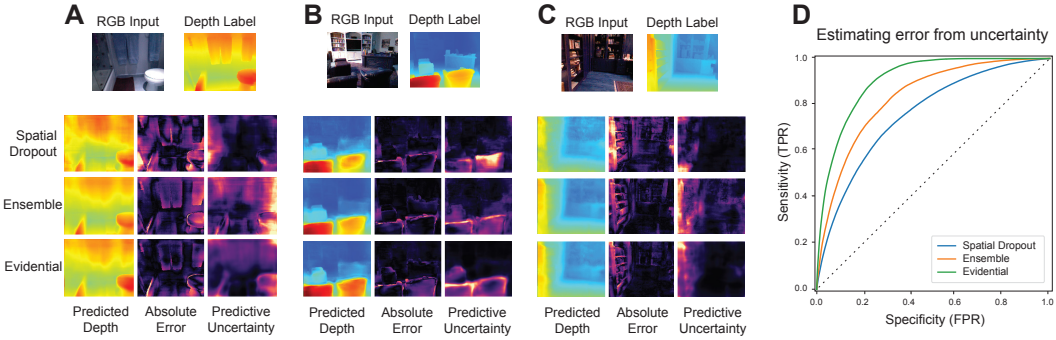

Figure 4: **Modeling uncertainty in depth estimation.** Three methods for estimating epistemic (model) uncertainty are evaluated in the context of monocular depth estimation. For each model, we visualize the prediction, error to ground-truth, and estimated uncertainty for three randomly chosen examples (A-C). Ideally, the model should predict high uncertainty whenever it does not know the answer (i.e., large error). We evaluate the sensitivity and specificity of the predictive uncertainty in identifying likely errors with ROC curves (D).

cf. Fig. 4. In general, dropout drastically underestimates the amount of uncertainty present, while ensembling occasionally overestimates the uncertainty, cf. Fig. 4A,C.

To further evaluate how calibrated the predictive uncertainty was to the ground-truth errors, we fit receiver operating characteristic (ROC) curves to normalized estimates of error and uncertainty. In other words, we test the network's ability to detect how likely it is to make an error at a given pixel using its predictive uncertainty. ROC curves take into account sensitivity and specificity of the uncertainties towards error predictions and are stronger if they contain greater area under their curve (AUC). Fig. 4D demonstrates that our evidential model provides uncertainty estimates which are the most attuned to where the model is making the errors.

4.3 Robustness to adversarial samples

A key use case of uncertainty estimation is to understand when a model is faced with test examples that fall outside of its training distribution or when the model's output cannot be trusted. In the previous subsection, we showed that our evidential uncertainties were well calibrated with the model's errors. In this subsection, we evaluate the uncertainty response for the depth estimation task under the extreme case where our model is presented with adversarially perturbed inputs.

We compute adversarial perturbations to our test set using the fast gradient sign method (Goodfellow et al., 2014), with increasing scales, ϵ, of noise. Fig. 5A confirms that the absolute error of all methods increasing as adversarial noise is added. We also observe a positive effect noise on our predictive uncertainty estimates in Fig. 5B. An additional desirable property of evidential uncertainty modeling is that it presents a higher overall uncertainty when presented with adversarial inputs compared to dropout and ensembling methods. Furthermore, we observe this strong overall uncertainty estimation despite the model losing calibration accuracy from the adversarial examples (Fig. 5C).

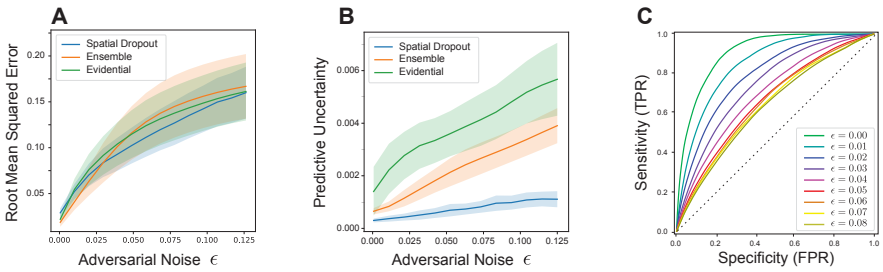

Figure 5: **Effect of adversarial examples.** Accuracy (A) and predictive uncertainty (B) are evaluated on various degrees of an adversarially perturbed testset. For evidential models, the relationship between adversarial noise level and ability of the estimate the error is also measured (C).

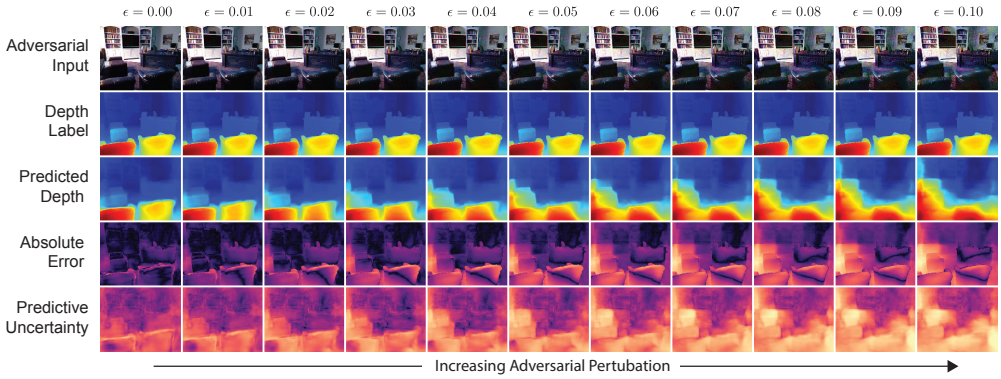

Figure 6: **Evidential robustness under adversarial noise.** Increasing levels of adversarial noise (left-to-right) corrupt the predicted depth, and our model begins to incur greater amounts of error. However, as adversarial noise increases, our evidence decreases and predictive uncertainty increases. Furthermore, the predictive uncertainty is localized to key areas where the error is increasing the most.

The robustness of evidential uncertainty against adversarial pertubations is visualized in greater detail in Fig. 6, which illustrates the predicted depth, error, and estimated pixel-wise uncertainty as we perturb the input image with greater amounts of noise (left-to-right). Note that the predictive uncertainty not only steadily increases as we increase the noise, but the spatial concentrations of uncertainty throughout the image maintain tight correspondence with the error.

5 Discussion and Related Work

Uncertainty estimation has had a long history in machine learning, from modeling probability distribution parameters using neural network outputs (Bishop, 1994) to Bayesian deep learning (Kendall & Gal, 2017). Our work builds on this foundation and presents a scalable representation for inferring the parameters of an evidential uncertainty distribution while simultaneously learning regression tasks via MLE.

In Bayesian deep learning, priors are placed over network weights and estimated using variational inference (Kingma et al., 2015). Dropout (Gal & Ghahramani, 2016; Molchanov et al., 2017) and Bayes-by-Backprop (Blundell et al., 2015) rely on multiple sampling iterations to estimate a predictive variance. Ensembles (Lakshminarayanan et al., 2017) provide a tangential approach where sampling occurs over multiple trained instances of the model. In contrast, we place uncertainty priors directly over our likelihood output function and thus only a single forward pass to evaluate both prediction and uncertainty. Additionally, our approach of uncertainty estimation proved to be better calibrated and capable of predicting where the model fails.

Our methodology falls under the class of evidential deep learning (Sensoy et al., 2018) which leverages the *Theory of Evidence* to model prior distributions and Dempster-Shafer theory (Khaleghi et al., 2013; Jsang, 2018) to interpret the resulting uncertainty. Prior work in this field Sensoy et al. (2018) has focused exclusively on modeling uncertainty in the classification domain with Dirichlet prior distributions. Our work extends this field into the broad range of regression learning tasks.

6 Conclusion

In this paper, we develop a novel method for training deterministic NNs that both estimates a desired target and evaluates the *evidence* in support of the target to generate robust metrics of model uncertainty. We formalize this in terms of learning evidential distributions, and achieve stable training by penalizing our model for prediction errors that scale with the available evidence. Our approach for evidential regression is validated on a benchmark regression task, and we further demonstrate that this method robustly scales to a key task in computer vision, depth estimation, and that the predictive uncertainty captured by our model tracks with adversarial input perturbations. This framework for evidential representation learning provides a means to achieve the precise uncertainty metrics required for robust neural ntwork deployment in safety-critical domains.

REFERENCES

Alexandre Alahi, Kratarth Goel, Vignesh Ramanathan, Alexandre Robicquet, Li Fei-Fei, and Silvio Savarese. Social lstm: Human trajectory prediction in crowded spaces. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 961–971, 2016.

Christopher M Bishop. Mixture density networks. 1994.

Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight uncertainty in neural networks. *arXiv preprint arXiv:1505.05424*, 2015.

Mariusz Bojarski, Davide Del Testa, Daniel Dworakowski, Bernhard Firner, Beat Flepp, Prasoon Goyal, Lawrence D Jackel, Mathew Monfort, Urs Muller, Jiakai Zhang, et al. End to end learning for self-driving cars. *arXiv preprint arXiv:1604.07316*, 2016.

Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pp. 1050–1059, 2016.

Timnit Gebru, Jonathan Krause, Yilun Wang, Duyun Chen, Jia Deng, Erez Lieberman Aiden, and Li Fei-Fei. Using deep learning and google street view to estimate the demographic makeup of neighborhoods across the united states. *Proceedings of the National Academy of Sciences*, 114 (50):13108–13113, 2017.

Clément Godard, Oisin Mac Aodha, and Gabriel J Brostow. Unsupervised monocular depth estimation with left-right consistency. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 270–279, 2017.

Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

Alex Graves. Practical variational inference for neural networks. In *Advances in neural information processing systems*, pp. 2348–2356, 2011.

José Miguel Hernández-Lobato and Ryan Adams. Probabilistic backpropagation for scalable learning of bayesian neural networks. In *International Conference on Machine Learning*, pp. 1861–1869, 2015.

Audun Jsang. *Subjective Logic: A formalism for reasoning under uncertainty*. Springer Publishing Company, Incorporated, 2018.

Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision? In *Advances in neural information processing systems*, pp. 5574–5584, 2017.

Bahador Khaleghi, Alaa Khamis, Fakhreddine O Karray, and Saiedeh N Razavi. Multisensor data fusion: A review of the state-of-the-art. *Information fusion*, 14(1):28–44, 2013.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Durk P Kingma, Tim Salimans, and Max Welling. Variational dropout and the local reparameterization trick. In *Advances in Neural Information Processing Systems*, pp. 2575–2583, 2015.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in Neural Information Processing Systems*, pp. 6402–6413, 2017.

Jingtuo Liu, Yafeng Deng, Tao Bai, Zhengping Wei, and Chang Huang. Targeting ultimate accuracy: Face recognition via deep embedding. *arXiv preprint arXiv:1506.07310*, 2015.

Dmitry Molchanov, Arsenii Ashukha, and Dmitry Vetrov. Variational dropout sparsifies deep neural networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 2498–2507. JMLR. org, 2017.

Pushmeet Kohli Nathan Silberman, Derek Hoiem and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *ECCV*, 2012.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pp. 234–241. Springer, 2015.

Murat Sensoy, Lance Kaplan, and Melih Kandemir. Evidential deep learning to quantify classification uncertainty. In *Advances in Neural Information Processing Systems*, pp. 3179–3189, 2018.

Joram Soch and Carsten Allefeld. Kullback-leibler divergence for the normal-gamma distribution. *arXiv preprint arXiv:1611.01437*, 2016.

Jonathan Tompson, Ross Goroshin, Arjun Jain, Yann LeCun, and Christoph Bregler. Efficient object localization using convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 648–656, 2015.

7 Appendix

7.1 Model evidence derivations

For convenience, define $\tau = 1/\sigma^2$ be the precision of a Gaussian distribution.

7.1.1 Type II Maximum Likelihood Loss

$$p(y|m) = \int_\tau \int_\mu p(y|\mu,\tau)\, p(\mu,\tau|\gamma,\lambda,\alpha,\beta)\ \mathrm{d}\mu\, \mathrm{d}\tau \tag{13}$$

$$= \int_{\tau=0}^{\infty} \int_{\mu=-\infty}^{\infty} \left[\sqrt{\frac{\tau}{2\pi}} e^{-\frac{\tau}{2}(y-\mu)^2}\right] \left[\frac{\beta^\alpha \sqrt{\lambda}}{\Gamma(\alpha)\sqrt{2\pi}} \tau^{\alpha-\frac{1}{2}} e^{-\beta\tau} e^{-\frac{\lambda\pi(\mu-\gamma)^2}{2}}\right] \mathrm{d}\mu\, \mathrm{d}\tau \tag{14}$$

$$= \int_{\tau=0}^{\infty} \frac{(\beta\tau)^\alpha}{\Gamma(\alpha)} \sqrt{\frac{\lambda}{2\pi\tau(1+\lambda)}} e^{-\beta\tau} e^{-\frac{\tau\lambda(\gamma-y)^2}{2(1-\lambda)}}\ \mathrm{d}\tau \tag{15}$$

$$= 2^{\frac{1}{2}+\alpha} \beta^\alpha \sqrt{\frac{\lambda}{2\pi(1+\lambda)}} \left(2\beta + \frac{\lambda(\gamma-y)^2}{1+\lambda}\right)^{-\frac{1}{2}-\alpha} \tag{16}$$

For computational reasons it is common to instead minimize the negative logarithm of the model evidence.

$$J = -\log p(y|m) = -\log\left(2^{\frac{1}{2}+\alpha} \beta^\alpha \sqrt{\frac{\lambda}{2\pi(1+\lambda)}} \left(2\beta + \frac{\lambda(\gamma-y)^2}{1+\lambda}\right)^{-\frac{1}{2}-\alpha}\right) \tag{17}$$

7.1.2 Sum of Squares Loss

$$\int_\tau \int_\mu \mathbb{E}_{y\sim p(y|\mu,\tau)}\left[||\hat{y}_i - y||_2^2\right] p(\mu,\tau|\gamma,\lambda,\alpha,\beta)\ \mathrm{d}y\, \mathrm{d}\mu\, \mathrm{d}\tau \tag{18}$$

$$= \int_\tau \int_\mu \int_y ||\hat{y}_i - y||_2^2 p(y|\mu,\tau)\, p(\mu,\tau|\gamma,\lambda,\alpha,\beta)\ \mathrm{d}y\, \mathrm{d}\mu\, \mathrm{d}\tau \tag{19}$$

$$= \int_{\tau=0}^{\infty} \int_{\mu=-\infty}^{\infty} \int_{y=-\infty}^{\infty} ||\hat{y}_i - y||_2^2 \left[\sqrt{\frac{\tau}{2\pi}} e^{-\frac{\tau}{2}(y-\mu)^2}\right] \tag{20}$$

$$\left[\frac{\beta^\alpha \sqrt{\lambda}}{\Gamma(\alpha)\sqrt{2\pi}} \tau^{\alpha-\frac{1}{2}} e^{-\beta\tau} e^{-\frac{\lambda\pi(\mu-\gamma)^2}{2}}\right] \mathrm{d}y\, \mathrm{d}\mu\, \mathrm{d}\tau \tag{21}$$

$$= \int_{\tau=0}^{\infty} \int_{\mu=-\infty}^{\infty} \left[(\hat{y}_i - \mu)^2 + \frac{1}{\tau}\right] \left[\frac{\beta^\alpha \sqrt{\lambda}}{\Gamma(\alpha)\sqrt{2\pi}} \tau^{\alpha-\frac{1}{2}} e^{-\beta\tau} e^{-\frac{\lambda\pi(\mu-\gamma)^2}{2}}\right] \mathrm{d}\mu\, \mathrm{d}\tau \tag{22}$$

$$J = -\log(\lambda) + \log[\beta(1+\lambda) + (\alpha-1)\lambda(y-\gamma)^2] + \log(\Gamma(\alpha-1)) - \log(\Gamma(\alpha)) \tag{23}$$