Epistemic Uncertainty Quantification For Pre-trained Neural Networks

Hanjing Wang
Rensselaer Polytechnic Institute
110 8th St, Troy, NY, 12180, USA
wangh36@rpi.edu

Qiang Ji
Rensselaer Polytechnic Institute
110 8th St, Troy, NY, 12180, USA
jiq@rpi.edu

Abstract

Epistemic uncertainty quantification (UQ) identifies where models lack knowledge. Traditional UQ methods, often based on Bayesian neural networks, are not suitable for pre-trained non-Bayesian models. Our study addresses quantifying epistemic uncertainty for any pre-trained model, which does not need the original training data or model modifications and can ensure broad applicability regardless of network architectures or training techniques. Specifically, we propose a gradient-based approach to assess epistemic uncertainty, analyzing the gradients of outputs relative to model parameters, and thereby indicating necessary model adjustments to accurately represent the inputs. We first explore theoretical guarantees of gradient-based methods for epistemic UQ, questioning the view that this uncertainty is only calculable through differences between multiple models. We further improve gradient-driven UQ by using class-specific weights for integrating gradients and emphasizing distinct contributions from neural network layers. Additionally, we enhance UQ accuracy by combining gradient and perturbation methods to refine the gradients. We evaluate our approach on out-of-distribution detection, uncertainty calibration, and active learning, demonstrating its superiority over current state-of-the-art UQ methods for pre-trained models.

1. Introduction

Uncertainty quantification (UQ) is essential in machine learning, especially as models tackle safety-critical tasks like healthcare diagnostics or autonomous navigation. UQ assesses predictive confidence, bolstering model trustworthiness and ensuring safer real-world applications.

There are two types of uncertainty: epistemic and aleatoric. Epistemic uncertainty stems from a lack of knowledge, often due to limited data or model inadequacies, and is potentially reducible given more training data. Aleatoric uncertainty arises from inherent randomness in the data and remains irreducible regardless of data availability. For classification tasks, aleatoric uncertainty in neural networks is often captured by the entropy of the softmax probability distribution. In contrast, Bayesian neural networks (BNNs) provide a systematic framework to estimate epistemic uncertainty by constructing the posterior distribution of model parameters. While the direct calculation of the posterior is often intractable, common methods include Markov Chain Monte Carlo [3, 9, 30, 34, 35, 38] and variational inference [4, 6, 14, 21, 23] techniques. BNNs then use samples from the posterior distribution to quantify epistemic uncertainty through prediction disagreements.

Our study highlights the importance of measuring epistemic uncertainty with any pre-trained model, avoiding the heavy computational efforts required by BNNs, and focusing on these advantageous conditions:

- **No Extra Knowledge Needed.** Our method works without requiring training or validation datasets, suitable for scenarios with limited data access.
- **No Model Refinement Required.** We avoid the need for parameter refinement, whether through retraining for new parameters or creating a parameter distribution from the trained model. This minimizes the computational demands and risks associated with model adjustments.
- **Universal Applicability.** UQ is universally applicable for any pre-trained model, free from constraints related to architecture design or specific training techniques.

Some existing methods such as gradient-based and perturbation-based UQ can meet the previously stated conditions. Gradient-based UQ is based on the idea that the sensitivity of a model's output to its parameters can indicate prediction uncertainties. Essentially, gradients show the adjustments needed for accurately representing the input, suggesting the model's insufficient knowledge about the input. Perturbation-based UQ, on the other hand, involves applying minor modifications to input data or model parameters to see how these changes impact predictions. If the model is unfamiliar with an input, even small perturbations can significantly change the output. In contrast, if the model knows the input well, it should also handle slight variations. Current gradient-based and perturbation-based methods have several limitations. Firstly, there's a lack of theoretical

analysis of their ability to accurately capture epistemic uncertainty. Additionally, their effectiveness often falls short compared to BNNs or even measures like softmax probability entropy [11]. Lastly, evaluations of these methods are primarily based on out-of-distribution detection, indicating a need for more comprehensive assessments.

Our study first challenges the idea that epistemic uncertainty is primarily detected through model discrepancies. Specifically, this paper offers theoretical support for general perturbation/gradient-based methods by connecting them to BNNs. We claim that gradient-based and perturbation-based methods are effective for this epistemic UQ with theoretical analysis, suggesting that a single model can recognize their limitations and areas of unfamiliarity. While conditions for ideal uncertainty estimation may not hold, we introduce three components to enhance gradient-based UQ based on the theoretical insights:

- **Class-specific Gradient Weighting.** In classification tasks, it's possible to calculate gradients for each class. We aim to develop a more effective uncertainty score by combining these gradients using class-specific weights.
- **Selective Gradient Computation.** We introduce layer-selective gradients, focusing on gradients from deeper layers of the neural network, as they are more representative of epistemic uncertainty.
- **Smoothing Noisy Gradients.** We combine gradients with perturbation techniques, applying slight input perturbations and aggregating gradients for each perturbed input to smooth the raw gradients.

2. Related Work

Gradient-based UQ. Gradient-based techniques leverage gradient information for uncertainty estimation. The key difference among gradient-based methods is their strategy for deriving an uncertainty score by computing and integrating gradients. For example, Lee et al. [19] backpropagated the cross entropy loss using the network's predicted probabilities with a confounding label. Huang et al [10] computed the gradients of the KL divergence between the predicted distribution and uniform distribution, where epistemic uncertainty is quantified via the norm of gradients. Meanwhile, Igoe et al. [11] introduced ExGrad, which scales the gradients of each class based on its associated predicted probability. While Lee et al. [19] offered intuitive reasoning, gradient-based UQ methods lack solid theoretical foundations, relying primarily on empirical results. As highlighted by Igoe et al. [11], these gradient-centric UQ methods often don't outperform simple measures like measuring the distance of the predicted distribution to a uniform one, which primarily captures aleatoric uncertainty. This motivates us to propose an improved approach for epistemic UQ, utilizing gradient information more effectively.

Perturbation-based UQ. Perturbation-based methods [1, 2, 18, 26] measure epistemic uncertainty by introducing small modifications to the inputs or parameters and observing the resulting variations in the outputs. Alarab et al. [1, 2] employed adversarial attacks to the inputs to estimate uncertainty. Ledda et al. [18] introduced extra dropout layers into a trained network that initially did not include dropout, utilizing these layers for epistemic UQ. Despite their empirical effectiveness, these methods lack theoretical support, motivating our theoretical analysis.

Other UQ Methods for Pre-trained Models. Contrary to perturbation-based and gradient-based techniques, the UQ methods outlined in this section mandate specific prerequisites in pre-trained models for effective epistemic UQ. Several methods are proposed to capture uncertainty through a special design of the network. For example, MC-dropout [7] assumes the network incorporates dropout layers during both training and testing for UQ. Similarly, MC-batchnorm [33] employs a deterministic network trained with batch normalization, which is also maintained during testing for UQ. Additionally, some methods require extra information beyond the model architecture. Laplace Approximation methods [4, 20, 22, 29] estimate the posterior distribution by approximating it as a Gaussian distribution and performing a Taylor expansion around the learned parameters of a pre-trained model. However, these methods necessitate access to the training data to approximate the posterior accurately. Utilizing the training data, Schweighofer et al. [31] explored the parameter space around a pre-trained model to identify alternative modes of the posterior distribution.

3. Theoretical Analysis

In this section, we first introduce the epistemic UQ of a pre-trained model. We then demonstrate how perturbation-based UQ methods can, under certain conditions, serve as effective approximations to BNNs for the estimation of epistemic uncertainty. Finally, we offer a theoretical analysis of the capacity of gradient-based techniques in quantifying epistemic uncertainty, highlighting the connections between perturbation-based and gradient-based approaches.

3.1. Epistemic Uncertainty Quantification

General Notations and Assumptions. Consider x as the input, y as the target variable, and $\mathcal{D}$ as the training data. We focus on classification tasks. Without loss of generality, we assume the pre-trained model is a single deterministic NN that provides a probability vector $p(y|x, \theta^*) \in [0, 1]^C$, where θ^* are the pre-determined model parameters and C is the class count. We use $\mathbb{E}(\cdot)$ for expectation, $\mathcal{H}(\cdot)$ for entropy, and $\mathcal{I}(\cdot;\cdot)$ to represent mutual information.

Epistemic UQ for BNNs. BNNs treat the neural network parameters, denoted as θ, as random variables governed by a posterior distribution $p(\theta|\mathcal{D})$. A widely used method to estimate epistemic uncertainty involves calculating the mu-

tual information between y and θ, expressed as:

$$\mathcal{I}(y;\theta|x,\mathcal{D}) = \mathbb{E}_{p(\theta|\mathcal{D})}\left[\mathrm{KL}(p(y|x,\theta)||p(y|x,\mathcal{D}))\right] \quad (1)$$

where KL represents the Kullback–Leibler divergence.

Desired Epistemic UQ for a Pre-trained Model. While the ideal Bayesian prediction $p(y|x,\mathcal{D})$ remains intractable, Schweighofer et al. [31] approximated it in Eq. (1) using $p(y|x,\theta^*)$ for quantifying epistemic uncertainty of a pre-trained model parameterized by θ^*. They aimed to estimate the prediction difference between θ^* and samples from the posterior distribution, which is formulated as follows:

$$U_e(x) = \mathbb{E}_{p(\theta|\mathcal{D})}\left[\mathrm{KL}(p(y|x,\theta)||p(y|x,\theta^*))\right]. \quad (2)$$

Considering $\theta = \theta^* + \Delta\theta$, we can reformulate $U_e(x)$ from Eq. (2) by performing perturbations on θ^*:

$$\mathbb{E}_{p(\Delta\theta|\mathcal{D})}\left[\mathrm{KL}(p(y|x,\theta^*+\Delta\theta)||p(y|x,\theta^*))\right]. \quad (3)$$

3.2. Perturbation-based UQ

In this section, we aim to explore the connection between perturbation-based UQ and BNNs for epistemic UQ. Proposition 3.1 provides a theoretical analysis indicating that, in certain conditions, perturbation-based methods can effectively approximate BNNs.

Proposition 3.1. *Assume the model parameters θ^* are learned given sufficient in-distribution training data $\mathcal{D}$, i.e., $|\mathcal{D}| \to \infty$. Under mild regularity conditions (i.e., the likelihood function of θ is continuous, θ^* is not on the boundary of the parameter space), perturbing θ^* by adding a small Gaussian noise can accurately approximate the posterior distribution $p(\theta|\mathcal{D})$. For example,*

$$\theta^* + \sigma\epsilon \sim \mathcal{N}(\theta^*, \sigma^2 I) \quad \textit{where} \quad \epsilon \sim \mathcal{N}(0, I) \quad (4)$$

and

$$\sup_{\theta} |p(\theta|\mathcal{D}) - \mathcal{N}(\theta;\theta^*,\sigma^2 I)| \to 0 \quad (5)$$

where $\sigma \to 0$ is a small positive constant.

The proof of Proposition 3.1 is based on the Bernstein-von Mises theorem [8, 13]. This theorem indicates that, given infinite training data, the posterior distribution tends toward a Gaussian distribution. Moreover, as the number of training data increases, this Gaussian distribution narrows. The detailed proof is shown in Appendix A. Proposition 3.1 underscores the potential of perturbation-based methods to approximate BNNs, especially in cases where the training data is comprehensive and the model is robustly trained. In this scenario, $\mathcal{N}(0,\sigma^2 I)$ closely approximates $p(\Delta\theta|\mathcal{D})$. It is worth noting that an adequately large dataset $\mathcal{D}$ in Eq. (5) makes diagonal covariance sufficient for reliable uncertainty measures. The assumption $|\mathcal{D}| \to \infty$ is significant as they (1) establish conditions for ideal uncertainty estimation and (2) enable the estimation of epistemic uncertainty for any pre-trained model in a challenging scenario, assuming no extra knowledge or learning is needed. While the assumptions are generally hard to achieve, their strict satisfaction isn't necessary when most datasets have sufficient samples and general data augmentation is applied during training. Furthermore, we can derive an upper bound for the total variation distance, $\mathrm{D_{TV}}(p(\theta|\mathcal{D}), N(\theta;\theta^*,\sigma^2 I))(\sigma \to 0)$, to analyze the validity of Proposition 3.1 for finite $\mathcal{D}$:

Proposition 3.2. *Denote $v(\theta) = -\frac{1}{|\mathcal{D}|}\log p(\theta) - \frac{1}{|\mathcal{D}|}\sum_{(x,y)\in\mathcal{D}} \log p(y|x,\theta)$ with $p(\theta)$ as a pre-defined prior distribution. Under Proposition 3.1 and the regularity constraints on $v(\theta)$ from Sec. 2.2 of [12], we have*

$$\begin{aligned} D_{TV} \leq & \sqrt{\frac{1}{2} KL(\mathcal{N}(\theta^*,\sigma^2 I)||\mathcal{N}(\theta^*,-H(\theta^*)^{-1})} \\ & + c(c_3^2(v) + c_4(v))\frac{d^2}{|\mathcal{D}|} + |c_3(v)|\frac{d}{\sqrt{|\mathcal{D}|}}. \end{aligned} \quad (6)$$

$H(\theta^*) = \nabla^2_{\theta^*}\log p(\theta^*|\mathcal{D})$. d is the dimension of θ, c is an absolute constant, and $c_3(v), c_4(v)$ are constants computed from third/fourth-order derivatives of v.

Note that $\mathcal{N}(\theta^*, -H(\theta^*)^{-1})$ is the Laplacian approximation of $p(\theta|\mathcal{D})$ and $H(\theta^*)^{-1} \to \mathbf{0}$ as $|\mathcal{D}| \to \infty$ based on the Bernstein-von Mises theorem. Hence, the upper bound $\to 0$ as $|\mathcal{D}| \to \infty$. It implies a good convergence when $|\mathcal{D}| \gg d^2$. The proof will be shown in Appendix A. While perturbations in the parameter space have a clear link to BNNs, input-space perturbations offer more understandable modifications. The large parameter space and complex nonlinear transformations in NNs make it challenging to identify effective parameter-space perturbations. This complexity also hinders the ability to understand how perturbations operate within neural networks. In contrast, applying perturbations directly to the input is relatively easier, and these perturbations can be visualized. Intuitively, when a model has limited knowledge about an input (indicative of high epistemic uncertainty), it probably also lacks knowledge about the nearby regions of that input. A slight perturbation to the input might induce a notable change in prediction. Additionally, the following proposition suggests parameter-space and input-space perturbations are transferable under certain conditions with proof shown in Appendix A:

Proposition 3.3. *Denote $f(x,\theta)$ as a neural network parameterized by θ with input x. For any small perturbation $\Delta\theta \to 0$, there exists a small perturbation Δx that fulfills*

$$f(x,\theta+\Delta\theta) = f(x+\Delta x,\theta). \quad (7)$$

Conversely, for any small input-space perturbation Δx, there exists a small perturbation applied on the first layer parameters that satisfies Eq. (7). Based on Eq. (7),

$$\begin{aligned} U_e(x) &= \mathbb{E}_{\Delta\theta}\left[\mathrm{KL}(p(y|x,\theta^*+\Delta\theta)||p(y|x,\theta^*))\right] \\ &= \mathbb{E}_{\Delta x}\left[\mathrm{KL}(p(y|x+\Delta x,\theta^*)||p(y|x,\theta^*))\right]. \end{aligned} \quad (8)$$

As a result, under the conditions of Proposition 3.1 and 3.3, suitable perturbations in the input space and subsequent evaluation of output variations can also effectively emulate the behavior of BNNs to measure epistemic uncertainty. Note that when both $\Delta\theta$ and Δx are small, Eq. (7) can be well-approximated by the first-order Taylor expansion of both sides. This reveals a linear relation between Δx and $\Delta\theta$. Thus, for small Gaussian $\Delta\theta$ as per Proposition. 3.1, Δx is also Gaussian and sampling $\Delta x \sim \mathcal{N}(0, \sigma^2 I)$ with a small σ is a suitable approximation.

3.3. Gradient-based UQ

Gradients tracing from the model's output back to its parameters are indicative of epistemic uncertainty. Fundamentally, these gradients measure the slight changes necessary in the model's parameters to better represent the input. When faced with an unfamiliar input, the model needs more substantial adjustments in its parameters. Conversely, for familiar, in-distribution samples, these gradients are close to zero. This concept is demonstrated in Proposition 3.4:

Proposition 3.4. *Let us assume the neural network f parameterized by θ has sufficient complexity and it is trained with a sufficiently large dataset $\mathcal{D}$ ($|\mathcal{D}| \to \infty$) in the neighborhood of an in-distribution input x. Under mild regularity conditions (i.e., the likelihood function of θ is continuous, θ^* is not on the boundary of the parameter space), the optimized θ^* achieves global optimality in x's neighborhood, denoted $\mathcal{N}(x)$. For $\forall x + \Delta x \in \mathcal{N}(x)$, we have:*

$$\frac{\partial f(x, \theta^*)}{\partial \theta^*} = 0 \qquad \frac{\partial f(x + \Delta x, \theta^*)}{\partial \theta^*} = 0 \qquad . \quad (9)$$

Proposition 3.4, whose proof is in Appendix A, theoretically supports using gradients as epistemic uncertainty indicators, as they may reflect data density and often approach zero for in-distribution data. Intuitively, out-of-distribution data often has non-zero gradients, as it is typically not well-represented by the model, suggesting a need for larger model adjustments to recognize them. Proposition 3.4 also indicates that the gradients around the neighborhood of in-distribution x are small. This inspires us to merge the gradients of neighboring samples with the original sample, creating smoother gradients to estimate epistemic uncertainty.

In exploring the relationship between perturbation-based and gradient-based UQ, it's evident that both approaches fundamentally depend on the sensitivity of the model's outputs to its parameters. While perturbation-based UQ directly modifies parameters or inputs and observes the resulting variations in outputs, gradient-based UQ achieves a similar understanding indirectly through gradient analysis. They do not actively change the parameters but employ gradients directly as a tool for constructing the uncertainty score. The gradients can be viewed as a first-order approximation of the changes observed in perturbation-based methods. Therefore, under small perturbations, both methods are likely to yield similar insights into the model's uncertainty. Building on the connection between perturbation-based and gradient-based methods, the effectiveness of perturbation-based methods by Proposition 3.1 in measuring epistemic uncertainty suggests a similar potential in gradient-based approaches. The subsequent proposition bridges gradient-based UQ with perturbation-based UQ mathematically:

Proposition 3.5. *The epistemic uncertainty derived by the expected gradient norm can serve as an upper bound compared to the uncertainty produced by perturbation-based methods when the perturbations are small.*

$$\begin{aligned}
&\mathbb{E}_{p(\Delta\theta)}\left[\mathrm{KL}(p(y|x,\theta^*)||p(y|x,\theta^* + \Delta\theta))\right] \\
&\leq \sum_{c=1}^{C} p(y = c|x, \theta^*) \left\| \frac{\partial \log p(y = c|x, \theta^*)}{\partial \theta^*} \right\| \mathbb{E}_{p(\Delta\theta)}[||\Delta\theta||] \\
&\propto \mathbb{E}_{y \sim p(y|x,\theta^*)} \left[\left\| \frac{\partial \log p(y|x, \theta^*)}{\partial \theta^*} \right\| \right] \text{(ExGrad [11])}
\end{aligned} \quad (10)$$

where $\Delta\theta \to 0$ and $\mathbb{E}_{p(\Delta\theta)}[||\Delta\theta||]$ is independent of x.

It is notable that the uncertainty expressed by perturbation-based methods in Eq. (10) employs the reverse KL divergence, in contrast to Eq. (2). Nevertheless, it remains a valid uncertainty metric. Proposition 3.5 further implies the necessity to adjust the gradients using the associated probability, indicating different treatments for the gradient of each class. Given the link between gradient-based UQ and perturbation-based techniques, under the conditions of Proposition 3.1, the epistemic uncertainty derived from gradient-based methods might also offer a good approximation to the epistemic uncertainty quantified by BNNs. Although the assumptions underlying these propositions might not always hold, it is noteworthy that ExGrad (as detailed in Eq.(10)) could surpass perturbation-based methods, especially in scenarios where perturbations significantly deviate from the optimum.

4. Proposed Method

Our proposed method, rooted in solid theoretical foundations, brings three key advancements to gradient-based UQ. First, as highlighted in Proposition 3.5, it underscores the necessity of assigning distinct weights to the gradients of each class. The proposed method aims to explore these weights to improve UQ effectiveness. Second, previous approaches uniformly apply the gradient to the parameters across all neural network layers, which may be unsuitable for practical applications. Lastly, motivated by Proposition 3.3, we combine input perturbations with gradients for

enhanced performance. Proposition 3.4 suggests leveraging gradients from samples near the target input when constructing the uncertainty score. By integrating gradients with perturbations, our approach seeks to enhance UQ by utilizing a smoother gradient representation.

4.1. Class-specific Gradient Weighting

When Eq. (10) suggests weighing the gradient of each class by its associated probability, the method can face challenges. It can produce overconfident results, where a high model probability overshadows the contributions from gradients of other classes, potentially leading to an overemphasis. This might neglect important insights about the model's decision boundaries and uncertain regions indicated by other class gradients. While the gradients of the class with the maximum probability still take the largest weight, we utilize the L2 probability-weighted gradient norm to normalize gradients with a square root:

$$U_{\text{REGrad}}(x) = \sum_{c=1}^{C} \sqrt{p(y=c|x,\theta^*) \left\| \frac{\partial \log p(y=c|x,\theta^*)}{\partial \theta^*} \right\|_2^2}. \tag{11}$$

The proposed L2 root-normalized expected gradient method (REGrad) effectively mitigates overconfidence issues, ensuring a more balanced incorporation of gradient information from all classes.

4.2. Layer-selective Gradients

In deep neural networks, each layer captures distinct feature representations, with initial layers extracting low-level features such as edges/texture and deeper layers interpreting complex patterns. Thus, the uncertainty at each layer varies, ranging from ambiguity in basic feature recognition to confusion in decision boundaries. Gradients from deeper layers excel at discerning classification patterns and may struggle to represent out-of-distribution samples, thus being more indicative of epistemic uncertainty.

Recognizing the hierarchical structure of neural networks, we introduce a weighting scheme that assigns importance to layers based on their depth as follows:

$$\left\| \frac{\partial \log p(y|x,\theta^*)}{\partial \theta^*} \right\| \xrightarrow[\text{selective}]{\text{layer}} \sum_{\theta_l^* \in \theta^*} a_l \left\| \frac{\partial \log p(y|x,\theta^*)}{\partial \theta_l^*} \right\|. \tag{12}$$

Based on Eq. (12), our approach does not treat all parameter gradients equally. Instead, we apply a coefficient a_l to the gradients of each layer θ_l, indexed by l. This coefficient progressively increases from early to deeper layers. Specifically, we use an exponential weighting mechanism, where the weight for layer l is defined as $a_l = \exp(\lambda l)$. Here, $\lambda > 0$ is a hyperparameter controlling the rate of exponential weight increase in deeper layers. By giving more weight to gradients from deeper layers, our method enhances the role of these gradients in measuring uncertainty. This approach emphasizes classification-specific insights and decision-making aspects crucial for detecting out-of-distribution instances. Notably, integrating these layer-selective gradients can improve any gradient-based UQ approach without adding computational complexity.

4.3. Gradient-Perturbation Integration

While gradients and perturbations are both valuable for understanding model behavior and uncertainty, we aim to combine their strengths for UQ. Our method calculates gradients for both the original and perturbed inputs to produce a smoother and more informative gradient representation, addressing the issue of noisy raw gradients. As Proposition 3.4 suggests, gradients from perturbed inputs offer additional uncertainty insights. This concept aligns with Smilkov et al.'s findings [32], where averaging gradients from perturbed inputs results in smoother gradients. While their research focused on using these gradients for clearer saliency maps in model explanation, we specifically apply these refined gradients to improve UQ. Specifically, for an input x^0, we introduce input perturbations $\Delta x \sim \mathcal{N}(0, \sigma^2 I)$, controlled by a hyperparameter σ. This process generates perturbed inputs $x^1, x^2, \cdots, x^N$. The smoothed gradients are then calculated as follows:

$$\left\| \frac{\partial \log p(y|x_0,\theta^*)}{\partial \theta^*} \right\| \xrightarrow[\text{smoothed}]{\text{perturb}} \left\| \frac{1}{N+1} \sum_{i=0}^{N} \frac{\partial \log p(y|x_i,\theta^*)}{\partial \theta^*} \right\| \tag{13}$$

Eq. (13) effectively mitigates sharp, uncharacteristic spikes in the gradient space, ensuring a more stable uncertainty measure. It is crucial to note that we maintain a small value for σ, aligning with the guidelines of Proposition 3.1. Calculating smoothed gradients does not complicate the backpropagation process, as it requires only one backward pass for the averaged probability vector. The additional computation mainly involves forward passes for the perturbed inputs. However, with all perturbed inputs prepared, these forward passes can be executed concurrently, leveraging parallel processing for efficiency.

5. Experiments

Dataset. We evaluated our method using benchmark image classification datasets, including MNIST [5], SVHN [27], CIFAR-10 (C10) [16], and CIFAR-100 (C100) [15].

Implementation Details. For MNIST and SVHN, we employed standard CNN architectures. For C10 and C100, we used ResNet18 and ResNet152, respectively. These models were trained following standard protocols to develop deterministic classification models, which then served as the pretrained models for our epistemic UQ experiments. Detailed experiment settings, baseline implementations, and hyperparameter information are provided in Appendix B.

Table 1. OOD detection results for AUROC (%) ↑ and AUPR (%) ↑ with epistemic uncertainty. "*" represents our method (REGrad + layer-selective + perturbation). The experiments are aggregated over three independent runs.

Method	MNIST → Omniglot		MNIST → FMNIST		C10→ SVHN		C10→ LSUN		Avg
	AUROC	AUPR	AUROC	AUPR	AUROC	AUPR	AUROC	AUPR	
NEGrad	41.99 ±1.75	43.49 ±0.91	40.99 ±3.51	44.41 ±0.91	72.80 ±1.27	64.47 ±1.09	66.96 ±2.05	58.51 ±1.64	54.20
UNGrad	92.81 ±0.09	90.79 ±0.33	95.41 ±0.48	94.91 ±0.34	38.07 ±9.79	46.86 ±9.31	17.08 ±8.56	33.52 ±9.27	63.68
GradNorm	16.52 ±1.47	35.47 ±0.61	22.71 ±2.01	40.88 ±0.61	10.73 ±3.09	32.04 ±1.15	10.57 ±3.93	32.04 ±0.89	25.12
Exgrad	97.55 ±0.05	96.99 ±0.16	98.11 ±0.14	97.98 ±0.16	88.87 ±0.01	84.85 ±1.15	88.23 ±0.11	82.26 ±1.02	91.86
Perturb x	97.16 ±0.00	95.65 ±0.16	97.09 ±0.14	95.39 ±0.31	**92.35** ±0.22	**91.57** ±1.32	82.89 ±0.40	72.35 ±0.55	90.56
Perturb θ	97.64 ±0.09	97.05 ±0.24	97.99 ±0.16	97.64 ±0.22	89.31 ±1.26	85.80 ±3.96	88.33 ±0.11	82.99 ±0.27	92.09
MC-AA	91.23 ±0.35	89.40 ±0.28	95.28 ±0.44	95.24 ±0.19	87.56 ±0.57	82.22 ±2.08	82.96 ±0.84	71.79 ±1.62	86.96
Inserted Dropout	97.52 ±0.14	96.88 ±0.35	97.00 ±0.68	95.75 ±0.79	88.84 ±1.02	88.01 ±1.19	88.18 ±0.62	86.93 ±0.73	92.39
Entropy	97.70 ±0.08	97.38 ±0.19	98.04 ±0.22	97.94 ±0.17	88.86 ±0.34	84.10 ±0.36	89.84 ±0.30	87.72 ±0.57	92.70
ExGrad V Term	97.70 ±0.02	97.39 ±0.13	98.04 ±0.14	97.95 ±0.26	89.04 ±0.43	84.64 ±0.41	89.96 ±0.18	88.09 ±0.29	92.85
LA	97.92 ±0.14	97.45 ±0.29	97.95 ±0.27	97.37 ±0.43	91.86 ±0.27	88.06 ±0.27	86.81 ±0.27	85.04 ±0.27	92.81
REGrad*	**98.19** ±0.02	**97.95** ±0.06	**98.78** ±0.14	**98.79** ±0.07	92.32 ±0.59	89.42 ±2.81	**91.11** ±0.44	**89.93** ±0.69	**94.56**

Method	SVHN → C10		SVHN → C100		C100→ SVHN		C100→ LSUN		Avg
	AUROC	AUPR	AUROC	AUPR	AUROC	AUPR	AUROC	AUPR	
NEGrad	30.26 ±1.80	39.13 ±1.31	30.93 ±1.24	39.38 ±0.98	44.62 ±2.65	42.52 ±1.96	43.99 ±2.21	42.74 ±3.28	39.20
UNGrad	68.28 ±3.45	69.21 ±3.18	68.34 ±2.77	68.95 ±2.42	44.34 ±7.36	43.44 ±6.66	47.46 ±4.63	45.41 ±5.36	56.93
GradNorm	24.49 ±3.75	36.56 ±2.06	25.62 ±2.72	37.26 ±1.42	10.67 ±2.08	32.11 ±0.44	27.53 ±3.91	36.97 ±2.19	28.90
Exgrad	88.57 ±0.61	85.55 ±1.33	88.25 ±0.62	85.21 ±1.14	79.99 ±1.37	72.68 ±0.82	76.93 ±1.64	71.00 ±2.62	81.02
Perturb x	72.76 ±0.36	60.60 ±0.26	73.96 ±0.42	62.27 ±0.25	86.49 ±1.62	86.76 ±1.72	41.10 ±1.42	44.35 ±0.72	66.04
Perturb θ	89.01 ±0.22	86.04 ±0.41	88.75 ±0.14	85.83 ±0.21	50.99 ±2.23	51.28 ±3.08	39.42 ±0.70	42.57 ±1.07	66.74
MC-AA	71.53 ±1.32	60.98 ±1.67	72.75 ±2.12	62.55 ±2.72	54.74 ±1.08	59.26 ±2.49	31.61 ±1.81	39.40 ±2.22	56.60
Inserted Dropout	88.47 ±1.23	83.97 ±0.71	86.96 ±1.09	82.40 ±0.88	57.39 ±1.46	55.49 ±2.02	59.41 ±1.61	58.30 ±0.92	71.55
Entropy	89.64 ±0.04	87.76 ±2.19	89.09 ±0.13	87.25 ±0.06	53.38 ±0.49	55.20 ±0.18	42.28 ±0.99	46.07 ±0.39	68.83
ExGrad V Term	89.72 ±0.08	87.90 ±1.87	89.19 ±0.07	87.40 ±0.02	50.35 ±0.42	52.66 ±0.73	41.67 ±1.51	45.91 ±1.12	68.10
LA	80.73 ±0.73	87.46 ±1.00	88.77 ±0.78	86.06 ±1.12	80.09 ±0.12	73.63 ±0.39	76.50 ±0.40	71.92 ±0.48	80.65
REGrad*	**91.11** ±0.19	**89.63** ±0.37	**90.37** ±0.26	**88.83** ±0.37	**88.06** ±1.29	**85.74** ±1.45	**79.11** ±0.32	**74.39** ±0.67	**85.91**

Baselines. Our method (REGrad + layer-selective + perturbation) is compared against various baselines:

- Gradient-based Methods:
  NEGrad [11] ($||\mathbb{E}_{y\sim p(y|x,\theta^*)}[\nabla_{\theta^*}\log p(y|x,\theta^*)]||$),
  UNGrad [11] ($\mathbb{E}_{y\sim\text{uniform}}[||\nabla_{\theta^*}\log p(y|x,\theta^*)||]$),
  GradNorm [10] ($||\nabla_{\theta^*}\mathbb{E}_{y\sim\text{uniform}}[\log p(y|x,\theta^*)]||$),
  ExGrad [11] ($\mathbb{E}_{y\sim p(y|x,\theta^*)}[||\nabla_{\theta^*}\log p(y|x,\theta^*)||]$).
- Perturbation-based Methods:
  Perturb x ($\mathbb{E}_{\Delta x\sim\mathcal{N}(0,\sigma^2 I)}$ $[\text{KL}(p(y|x+\Delta x,\theta^*)||p(y|x,\theta^*))]$),
  Perturb θ ($\mathbb{E}_{\Delta\theta\sim\mathcal{N}(0,\sigma^2 I)}$ $[\text{KL}(p(y|x,\theta^*+\Delta\theta)||p(y|x,\theta^*))]$),
  MC-AA [1], Inserted Dropout [18].
- Entropy-based Methods:
  Entropy (-$\mathbb{E}_{y\sim p(y|x,\theta^*)}[\log p(y|x,\theta^*)]$),
  ExGrad V Term ($\sum_c |p(y=c|x,\theta^*)-\frac{1}{C}|$)
- Approximated BNN using training data: LA [14].

Evaluation Tasks. Section 5.1 presents our evaluation of epistemic UQ performance in out-of-distribution (OOD) detection. Section 5.2 details our uncertainty calibration evaluations. In Section 5.3, we discuss our findings from uncertainty-guided active learning evaluations.

5.1. Out-of-distribution Detection

OOD detection is a crucial application for UQ [24, 25]. It aims to identify data points that deviate from the training data distribution by utilizing uncertainty measures. Epistemic uncertainty, inversely correlating with data density, tends to be higher when the model encounters anomalous data.

Experiment Settings. For MNIST dataset, OOD samples come from the Omniglot [17] and Fashion-MNIST (FMNIST) [36] datasets. In the case of C10 and C100, the OOD datasets are SVHN [27] and LSUN [37]. Conversely, for SVHN as in-distribution data, C10 and C100 serve as OOD datasets. We evaluated OOD detection using two metrics: area under the receiver operating characteristic curve (AUROC) and area under the precision-recall curve (AUPR).

Experiment Analysis In the OOD detection experiments shown in Table 1, REGrad demonstrates superior performance over various baselines, including gradient-based, perturbation-based, and entropy-based methods. It achieves an impressive average improvement of 30% to 60% over NEGrad, UNGrad, and GradNorm. Moreover, REGrad* consistently outperforms the strongest competitor in each category. Against ExGrad, the state-of-the-art gradient-based method, REGrad* shows average enhancements of 2.7% and 4.84% across two tables. Within the perturbation-based category, it notably surpasses Perturb x and Perturb θ by 19.82% and 19.12% for SVHN and C100 datasets,

Table 2. Uncertainty calibration results for MNIST, C10, SVHN, and C100 datasets using rAULC ↑. "*" represents our method and the experiments are aggregated over three independent runs.

Method	MNIST	C10	SVHN	C100	Avg
	rAULC	rAULC	rAULC	rAULC	
NEGrad	0.143 ±.055	0.598 ±.039	-0.861 ±.042	0.127 ±.050	0.002
UNGrad	0.934 ±.004	0.840 ±.022	0.429 ±.061	0.403 ±.038	0.652
GradNorm	-2.810 ±.022	-1.275 ±.095	-1.176 ±.026	-0.995 ±.098	-1.564
ExGrad	**0.985** ±.001	0.893 ±.016	0.868 ±.003	0.841 ±.017	0.897
Perturb x	**0.985** ±.001	0.890 ±.014	0.830 ±.001	0.772 ±.032	0.869
Perturb θ	0.984 ±.001	0.892 ±.015	0.869 ±.001	0.705 ±.002	0.863
MC-AA	0.969 ±.002	0.879 ±.018	0.807 ±.000	0.784 ±.017	0.860
Inserted Dropout	0.973 ±.003	0.891 ±.005	0.851 ±.006	0.783 ±.003	0.874
Entropy	**0.985** ±.001	0.893 ±.015	0.867 ±.004	**0.863** ±.017	0.902
ExGrad V Term	**0.985** ±.001	0.893 ±.015	0.865 ±.004	0.854 ±.020	0.899
LA	0.983 ±.002	0.865 ±.003	0.857 ±.002	0.792 ±.007	0.874
REGrad*	**0.985** ±.001	**0.898** ±.016	**0.873** ±.002	0.858 ±.013	**0.904**

respectively. Compared to entropy-based methods, which mainly capture aleatoric uncertainty, REGrad*'s improvement over Entropy and ExGrad V Term ranged from 1.71% to 17.76% in averaged results across both tables. Even against LA, which approximates the posterior distribution of parameters using training data, REGrad* shows considerable improvements. Notably, given its assumption of a single Gaussian posterior and the computational constraints requiring a diagonal or block-diagonal covariance matrix, LA may not always be the most effective. These findings underscore REGrad*'s robustness and advanced capability in epistemic UQ across diverse scenarios.

5.2. Uncertainty Calibration

In this section, we demonstrate the effectiveness of our methods in uncertainty calibration. Uncertainty calibration measures the alignment of a model's predicted uncertainty with its actual performance. Well-calibrated uncertainty is crucial for informed decision-making, as it indicates not just what a model predicts, but also the reliability of those predictions. Traditional calibration metrics, such as the expected calibration error and negative log-likelihood, are unsuitable for our scenario. These metrics typically assess $p(y|x, \theta^*)$, but in our case, $p(y|x, \theta^*)$ remains constant across all UQ baselines when using a pre-trained model θ^*. Therefore, we opt for the relative area under the lift curve (rAULC) as our evaluation metric. Introduced in [28], rAULC is derived by ordering predictions by increasing uncertainty, then plotting the performance for samples with an uncertainty value below a certain quantile of the uncertainty against that quantile.

Experiment Analysis The uncertainty calibration results in Table 2 show that REGrad* consistently outperforms perturbation-based methods and LA, achieving approximately 3% to 4% higher performance. Moreover, our method significantly surpasses most gradient-based methods, such as NEGrad, UNGrad, and GradNorm. However, it exhibits only a marginal improvement over ExGrad. ExGrad's effectiveness in uncertainty calibration may be due to its use of the predictive probability vector for weighting gradients of each class. This suggests that the probability vector could be a reliable indicator of model performance. Additionally, REGrad* shows only a slight advantage over Entropy and ExGrad V Term. This can be attributed to the fact that the well-measured aleatoric uncertainty can also indicate model performance, as noted in [28]. Overall, REGrad* serves as the most effective performance indicator among all evaluated baselines.

5.3. Active Learning

Epistemic uncertainty is important in active learning (AL), where the aim is to annotate data points with the highest uncertainty for retraining the model, thus targeting areas where the model has minimal information. Initially, we train the model with m_1 data points. Over 10 acquisition cycles, we select m_2 new data points at each cycle from the unused training pool, focusing on those with the highest epistemic uncertainty to retrain the model along with the previously chosen data. Our experiments include MNIST, C10, SVHN, and C100 datasets, setting m_1 at 20, 500, 500, and 1000, and m_2 at 20, 100, 100, and 500 for each dataset, respectively. Due to the experiments' reliance on smaller data subsets, we use a standard CNN with two convolutional and two fully connected layers across all datasets. Implementation details are available in Appendix B. We evaluate the model's performance by measuring accuracy (ACC) and negative log-likelihood (NLL) on the original testing data.

Experiment Results. Table 3 presents the average ACC and NLL across 10 active learning acquisition cycles. Visualizations that illustrate the ACC/NLL progression for all datasets throughout these cycles can be found in Appendix C. REGrad* demonstrates superior performance over various baselines. For example, it shows a 1.72% improvement in ACC over ExGrad. Compared to perturbation-based

Table 3. Active learning results of averaged ACC ↑ and NLL ↓ across 10 acquisition cycles for all datasets . "*" represents our method.

Method	MNIST		C10		SVHN		C100		Avg	
	ACC	NLL	ACC	NLL	ACC	NLL	ACC	NLL	ACC	NLL
NEGrad	70.22 ±3.02	1.833 ±.04	39.13 ±.72	2.074 ±.02	65.12 ±2.25	1.827 ±.02	13.16 ±.27	4.507 ±.01	46.91	2.560
UNGrad	69.56 ±2.94	1.867 ±.03	38.90 ±.74	2.071 ±.01	67.27 ±2.17	1.808 ±.02	12.56 ±.58	4.508 ±.01	47.07	2.564
GradNorm	66.85 ±3.18	1.884 ±.04	39.02 ±.81	2.069 ±.01	65.43 ±2.30	1.827 ±.03	12.89 ±.33	4.509 ±.01	46.05	2.572
ExGrad	69.91 ±2.21	1.847 ±.03	39.20 ±.87	2.069±.01	66.76 ±2.11	1.812 ±.02	12.55 ±.32	4.506 ±.00	47.11	2.559
Perturb x	71.10 ±3.43	1.844 ±.05	38.57 ±.72	2.072±.01	64.00 ±2.39	1.838 ±.02	12.23 ±.27	4.511 ±.00	46.47	2.566
Perturb θ	71.44 ±2.43	1.826 ±.03	38.97 ±1.03	2.070 ±.02	66.31 ±2.62	1.817 ±.03	13.17 ±.37	4.506 ±.01	47.47	2.555
MC-AA	71.69 ±.42	1.809 ±.00	38.57 ±.72	2.073 ±.01	64.56 ±1.17	1.831 ±.02	12.14 ±.40	4.515 ±.01	46.74	2.557
Inserted Dropout	72.84 ±0.62	1.836 ±.00	38.34 ±.11	2.075 ±.01	66.99 ±2.58	1.811 ±.02	12.59 ±.13	4.507 ±.00	47.69	2.557
Entropy	70.40 ±4.33	1.846 ±.06	38.73 ±.65	2.074 ±.01	63.42 ±2.86	1.845 ±.03	11.57 ±.41	4.521 ±.01	46.03	2.572
ExGrad V Term	68.53 ±4.40	1.881 ±.04	38.66 ±.67	2.071 ±.01	62.32 ±2.87	1.856 ±.03	11.30 ±.38	4.520 ±.01	45.20	2.582
LA	62.65 ±.58	1.935 ±.02	37.56 ±.29	2.083 ±.00	63.44 ±.75	1.846 ±.00	12.59 ±.12	**4.505** ±.00	44.06	2.592
REGrad*	**75.31** ±3.48	**1.808** ±.03	**39.28** ±.83	**2.067** ±.01	**67.37** ±2.13	**1.807** ±.02	**13.37** ±.25	**4.505** ±.00	**48.83**	**2.547**

methods, REGrad* outperforms Perturb x with a 2.36% increase in ACC. Moreover, compared to LA, REGrad* achieves significant enhancements in both ACC and NLL. These improvements are particularly noteworthy given that training data is limited.

5.4. Ablation Studies

Effectiveness of Component Contributions. This section delves into the individual impact of REGrad, layer-selective gradients, and perturbations in OOD detection. The detailed results are presented in Table 4. It shows that REGrad itself can achieve enhanced performance compared to ExGrad, with further improvements observed when layer-selective gradients are added. The combination of REGrad with both layer-selective gradients and perturbations yields the best results, indicating the cumulative positive impact of these components. More analysis is shown in Appendix D.

Table 4. OOD detection for ablation studies using AUROC/AUPR.

Method	MNIST → FMNIST	C10 → SVHN
ExGrad	98.11/97.98	88.87/84.85
REGrad	98.42/98.45	89.64/85.75
REGrad + layer-selective	98.51/98.55	90.33/87.45
REGrad + layer-selective + perturb	**98.78/98.79**	**92.32/89.42**

Method	SVHN → C10	C100 → SVHN
ExGrad	88.57/85.55	79.99/72.68
REGrad	90.22/88.16	81.43/81.12
REGrad + layer-selective	90.72/88.82	87.32/83.39
REGrad + layer-selective + perturb	**91.11/89.63**	**88.06/85.74**

Hyperparameter Analysis. We analyze the coefficient λ in layer-selective gradients which adjusts weights towards deeper layers. As $\lambda \to \infty$, only the last layer's parameters influence gradient computation, whereas $\lambda \to 0$ treats all parameters equally. Our study in Table 5 shows that the OOD detection performance is relatively stable within specific λ ranges. Additionally, we examine the impact of σ in perturbations, selecting small values as per Proposition 3.1, and find the results are similarly stable within certain σ ranges. More analysis is shown in Appendix D.

Efficiency Analysis. While the complexity of the layer-selective gradients and the perturbation-integrated gradients are discussed in Section 4, we provide the empirical runtime in Appendix D.

Table 5. OOD detection for hyperparameter analysis using AUROC/AUPR. We use $\lambda = 0.3, \sigma = 0.02$ for our method.

Method	SVHN→ C10
REGrad + layer-selective ($\lambda \to 0$)	90.22/88.16
REGrad + layer-selective ($\lambda \to \infty$)	88.21/83.92
REGrad + layer-selective ($\lambda = 0.25$)	90.65/88.72
REGrad + layer-selective ($\lambda = 0.3$)	90.72/88.82
REGrad + layer-selective ($\lambda = 0.35$)	90.80/88.82
REGrad + layer-selective ($\lambda = 0.3$) + perturb ($\sigma = .015$)	91.05/89.55
REGrad + layer-selective ($\lambda = 0.3$) + perturb ($\sigma = .02$)	**91.11/89.63**
REGrad + layer-selective ($\lambda = 0.3$) + perturb ($\sigma = .025$)	90.98/89.41

Varying Model Architecture. While the main evaluation focuses on pre-trained CNN-based models, we also assess our method's effectiveness on vision transformers in Appendix D, where it continues to outperform other baselines.

6. Conclusion

In our study, we present a novel gradient-based method for epistemic UQ in pre-trained models, beginning with a theoretical analysis of gradient-based and perturbation-based methods' capabilities in capturing epistemic uncertainty. To improve current gradient-based methods, we introduce class-specific gradient weighting, layer-selective gradients, and gradient-perturbation integration. The proposed method does not require original training data or model modifications, ensuring broad applicability across any architecture and training technique. Our experiments across diverse scenarios, including out-of-distribution detection, uncertainty calibration, and active learning, demonstrate the superiority of our method over current state-of-the-art UQ methods for pre-trained models.

Acknowledgement. This work is supported in part by the National Science Foundation award IIS 2236026.

References

[1] Ismail Alarab and Simant Prakoonwit. Adversarial attack for uncertainty estimation: identifying critical regions in neural networks. *Neural Processing Letters*, 54(3):1805–1821, 2022.

[2] Ismail Alarab and Simant Prakoonwit. Uncertainty estimation based adversarial attack in multi-class classification. *Multimedia Tools and Applications*, 82(1):1519–1536, 2023.

[3] Tianqi Chen, Emily B. Fox, and Carlos Guestrin. Stochastic gradient hamiltonian monte carlo. In *Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32*, page II–1683–II–1691. JMLR.org, 2014.

[4] Erik Daxberger, Eric Nalisnick, James U Allingham, Javier Antorán, and José Miguel Hernández-Lobato. Bayesian deep learning via subnetwork inference. In *International Conference on Machine Learning*, pages 2510–2521. PMLR, 2021.

[5] Li Deng. The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012.

[6] Gianni Franchi, Andrei Bursuc, Emanuel Aldea, Séverine Dubuisson, and Isabelle Bloch. Encoding the latent posterior of bayesian neural networks for uncertainty quantification. *arXiv preprint arXiv:2012.02818*, 2020.

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

[8] Andrew Gelman. Induction and deduction in bayesian data analysis. 2011.

[9] Stuart Geman and Donald Geman. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Transactions on pattern analysis and machine intelligence*, (6):721–741, 1984.

[10] Rui Huang, Andrew Geng, and Yixuan Li. On the importance of gradients for detecting distributional shifts in the wild. *Advances in Neural Information Processing Systems*, 34:677–689, 2021.

[11] Conor Igoe, Youngseog Chung, Ian Char, and Jeff Schneider. How useful are gradients for ood detection really? *arXiv preprint arXiv:2205.10439*, 2022.

[12] Anya Katsevich. Tight dimension dependence of the laplace approximation. *arXiv preprint arXiv:2305.17604*, 2023.

[13] Bas JK Kleijn and Aad W van der Vaart. The bernstein-von-mises theorem under misspecification. *Electronic Journal of Statistics*, 6:354–381, 2012.

[14] Agustinus Kristiadi, Matthias Hein, and Philipp Hennig. Being bayesian, even just a bit, fixes overconfidence in relu networks. In *International Conference on Machine Learning*, pages 5436–5446. PMLR, 2020.

[15] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[16] Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. The cifar-10 dataset. *online: http://www. cs. toronto. edu/kriz/cifar. html*, 55(5), 2014.

[17] Brenden M Lake, Ruslan Salakhutdinov, and Joshua B Tenenbaum. Human-level concept learning through probabilistic program induction. *Science*, 350(6266):1332–1338, 2015.

[18] Emanuele Ledda, Giorgio Fumera, and Fabio Roli. Dropout injection at test time for post hoc uncertainty quantification in neural networks. *arXiv preprint arXiv:2302.02924*, 2023.

[19] Jinsol Lee and Ghassan AlRegib. Gradients as a measure of uncertainty in neural networks. In *2020 IEEE International Conference on Image Processing (ICIP)*, pages 2416–2420. IEEE, 2020.

[20] Jongseok Lee, Matthias Humt, Jianxiang Feng, and Rudolph Triebel. Estimating model uncertainty of neural networks in sparse information form. In *International Conference on Machine Learning*, pages 5702–5713. PMLR, 2020.

[21] Christos Louizos and Max Welling. Multiplicative normalizing flows for variational bayesian neural networks. In *International Conference on Machine Learning*, pages 2218–2227. PMLR, 2017.

[22] David J. C. MacKay. A Practical Bayesian Framework for Backpropagation Networks. *Neural Computation*, 4(3):448–472, 1992.

[23] Wesley J Maddox, Pavel Izmailov, Timur Garipov, Dmitry P Vetrov, and Andrew Gordon Wilson. A simple baseline for bayesian uncertainty in deep learning. In *Advances in Neural Information Processing Systems 32*, pages 13132–13143. Curran Associates, Inc., 2019.

[24] Andrey Malinin and Mark Gales. Predictive uncertainty estimation via prior networks. In *Advances in Neural Information Processing Systems 31*, pages 7047–7058. Curran Associates, Inc., 2018.

[25] Andrey Malinin, Bruno Mlodozeniec, and Mark Gales. Ensemble distribution distillation. *arXiv preprint arXiv:1905.00076*, 2019.

[26] Lu Mi, Hao Wang, Yonglong Tian, Hao He, and Nir N Shavit. Training-free uncertainty estimation for dense regression: Sensitivity as a surrogate. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 10042–10050, 2022.

[27] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. 2011.

[28] Janis Postels, Mattia Segu, Tao Sun, Luc Van Gool, Fisher Yu, and Federico Tombari. On the practicality of deterministic epistemic uncertainty. *arXiv preprint arXiv:2107.00649*, 2021.

[29] Hippolyt Ritter, Aleksandar Botev, and David Barber. A scalable laplace approximation for neural networks. In *6th International Conference on Learning Representations, ICLR 2018 - Conference Track Proceedings*, 2018.

[30] Christian P Robert, George Casella, and George Casella. *Monte Carlo statistical methods*. Springer, 1999.

[31] Kajetan Schweighofer, Lukas Aichberger, Mykyta Ielanskyi, Günter Klambauer, and Sepp Hochreiter. Quantification of uncertainty with adversarial models. *arXiv preprint arXiv:2307.03217*, 2023.

[32] Daniel Smilkov, Nikhil Thorat, Been Kim, Fernanda Viégas, and Martin Wattenberg. Smoothgrad: removing noise by adding noise. *arXiv preprint arXiv:1706.03825*, 2017.

[33] Mattias Teye, Hossein Azizpour, and Kevin Smith. Bayesian uncertainty estimation for batch normalized deep networks. In *International Conference on Machine Learning*, pages 4907–4916. PMLR, 2018.

[34] Luke Tierney. Markov chains for exploring posterior distributions. *the Annals of Statistics*, pages 1701–1728, 1994.

[35] Max Welling and Yee W Teh. Bayesian learning via stochastic gradient langevin dynamics. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pages 681–688, 2011.

[36] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

[37] Fisher Yu, Yinda Zhang, Shuran Song, Ari Seff, and Jianxiong Xiao. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. *CoRR*, abs/1506.03365, 2015.

[38] Ruqi Zhang, Chunyuan Li, Jianyi Zhang, Changyou Chen, and Andrew Gordon Wilson. Cyclical stochastic gradient mcmc for bayesian deep learning. *arXiv preprint arXiv:1902.03932*, 2019.