# Gradient-based Uncertainty Attribution for Explainable Bayesian Deep Learning

Hanjing Wang
Rensselaer Polytechnic Institute
wangh36@rpi.edu

Dhiraj Joshi
IBM Research
djoshi@us.ibm.com

Shiqiang Wang
IBM Research
wangshiq@us.ibm.com

Qiang Ji
Rensselaer Polytechnic Institute
jiq@rpi.edu

## Abstract

*Predictions made by deep learning models are prone to data perturbations, adversarial attacks, and out-of-distribution inputs. To build a trusted AI system, it is therefore critical to accurately quantify the prediction uncertainties. While current efforts focus on improving uncertainty quantification accuracy and efficiency, there is a need to identify uncertainty sources and take actions to mitigate their effects on predictions. Therefore, we propose to develop explainable and actionable Bayesian deep learning methods to not only perform accurate uncertainty quantification but also explain the uncertainties, identify their sources, and propose strategies to mitigate the uncertainty impacts. Specifically, we introduce a gradient-based uncertainty attribution method to identify the most problematic regions of the input that contribute to the prediction uncertainty. Compared to existing methods, the proposed UA-Backprop has competitive accuracy, relaxed assumptions, and high efficiency. Moreover, we propose an uncertainty mitigation strategy that leverages the attribution results as attention to further improve the model performance. Both qualitative and quantitative evaluations are conducted to demonstrate the effectiveness of our proposed methods.*

## 1. Introduction

Despite significant progress in many fields, conventional deep learning models cannot effectively quantify their prediction uncertainties, resulting in overconfidence in unknown areas and the inability to detect attacks caused by data perturbations and out-of-distribution inputs. Left unaddressed, this may cause disastrous consequences for safety-critical applications, and lead to untrustworthy AI models.

The predictive uncertainty can be divided into epistemic uncertainty and aleatoric uncertainty [16]. Epistemic uncertainty reflects the model's lack of knowledge about the input. High epistemic uncertainty arises in regions, where there are few or no observations. Aleatoric uncertainty measures the inherent stochasticity in the data. Inputs with high noise are expected to have high aleatoric uncertainty. Conventional deep learning models, such as deterministic classification models that output softmax probabilities, can only estimate the aleatoric uncertainty.

Bayesian deep learning (BDL) offers a principled framework for estimating both aleatoric and epistemic uncertainties. Unlike the traditional point-estimated models, BDL constructs the posterior distribution of model parameters. By sampling predictions from various models derived from the parameter posterior, BDL avoids overfitting and allows for systematic quantification of predictive uncertainties. However, current BDL methods primarily concentrate on enhancing the accuracy and efficiency of uncertainty quantification, while failing to explicate the precise locations of the input data that cause predictive uncertainties and take suitable measures to reduce the effects of uncertainties on model predictions.

Uncertainty attribution (UA) aims to generate an uncertainty map of the input data to identify the most problematic regions that contribute to the prediction uncertainty. It evaluates the contribution of each pixel to the uncertainty, thereby increasing the transparency and interpretability of BDL models. Previous attribution methods are mainly developed for classification attribution (CA) with deterministic neural networks (NNs) to find the contribution of image pixels to the classification score. Unlike UA, directly leveraging the gradient-based CA methods for detecting problematic regions is unreliable. While CA explains the model's classification process, assuming its predictions are confident, UA intends to identify the sources of input imperfections that contribute to the high predictive uncertainties. Moreover, CA methods are often class-discriminative since the classification score depends on the predicted class. As a result, they often fail to explain the inputs which have wrong predictions with large uncertainty [28]. Also shown by Ancona et al. [1], they are not able to show the troublesome areas of images for complex datasets. Existing CA methods can be categorized into gradient-based methods [15, 31, 33–37, 41, 43] and perturbation-based methods [7, 10, 11, 29, 30, 42]. The former directly utilizes the gradient information as input attribution, while the latter modifies the input and observes the corresponding output change. However, perturbation-based methods often require thousands of forward propagations to attribute one image, suffering from high complexity and attribution performance varies for different chosen perturbations. Although CA methods are not directly applicable, we will discuss their plain extensions for uncertainty attribution in Sec. 2.2.

Recently, some methods are specifically proposed for UA. For example, CLUE [3] and its variants [20, 21] aim at generating a better image with minimal uncertainty by modifying the uncertain input through a generative model, where the attribution map is generated by measuring the difference between the original input and the modified input. Perez et al. [28] further combine CLUE with the path integral for improved pixel-wise attributions. However, these methods are inefficient for real-time applications because they require solving one optimization problem per input for a modified image. Moreover, training generative models is generally hard and can be unreliable for complex tasks.

We propose a novel gradient-based UA method, named UA-Backprop, to effectively address the limitations of existing methods. The contributions are summarized below.

- UA-Backprop backpropagates the uncertainty score to the pixel-wise attributions, without requiring a pre-trained generative model or additional optimization. The uncertainty is fully attributed to satisfy the completeness property, i.e., the uncertainty can be decomposed into the sum of individual pixel attributions. The explanations can be generated efficiently within a single backward pass of the BDL model.
- We introduce an uncertainty mitigation approach that employs the produced uncertainty map as an attention mechanism to enhance the model's performance. We present both qualitative and quantitative evaluations to validate the efficacy of our proposed method.

## 2. Preliminaries

### 2.1. BDL and Uncertainty Quantification

BDL models assume that the neural network parameters $\boldsymbol{\theta}$ are random variables, with a prior $p(\boldsymbol{\theta})$ and a likelihood $p(\mathcal{D}|\boldsymbol{\theta})$, where $\mathcal{D}$ represents the training data. We can apply the Bayes' rule to compute the posterior of $\boldsymbol{\theta}$, i.e., $p(\boldsymbol{\theta}|\mathcal{D})$ as shown in the following equation:

$$p(\boldsymbol{\theta} \mid \mathcal{D}) = \frac{p(\mathcal{D} \mid \boldsymbol{\theta})p(\boldsymbol{\theta})}{p(\mathcal{D})}. \tag{1}$$

Computing the posterior analytically is often intractable. Therefore, various methods have been proposed for approximately generating parameter samples from the posterior, including MCMC sampling methods [6, 12, 13], variational methods [5, 22–24], and ensemble-based methods [14,19,38–40]. The advantages of the BDL models are their capability to quantify aleatoric and epistemic uncertainties.

Let us denote the input as $\boldsymbol{x}$, the target variable as $\boldsymbol{y}$, and the output target distribution as $p(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{\theta})$ parameterized by $\boldsymbol{\theta}$, which are the Bayesian parameters such that $\boldsymbol{\theta} \sim p(\boldsymbol{\theta}|\mathcal{D})$. In this paper, we will focus on classification tasks. For a given input $\boldsymbol{x}$ and training data $\mathcal{D}$, we estimate the epistemic uncertainty and the aleatoric uncertainty by the mutual information and the expected entropy [9] in:

$$\underbrace{\mathcal{H}[p(\boldsymbol{y}|\boldsymbol{x},\mathcal{D})]}_{\text{Total Uncertainty } U_t} = \underbrace{\mathcal{I}[\boldsymbol{y},\boldsymbol{\theta}|\boldsymbol{x},\mathcal{D}]}_{\text{Epistemic Uncertainty } U_e} + \underbrace{\mathbb{E}_{p(\boldsymbol{\theta}|\mathcal{D})}[\mathcal{H}[p(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\theta})]]}_{\text{Aleatoric Uncertainty } U_a} \tag{2}$$

where $\mathcal{H}$, $\mathcal{I}$, and $\mathbb{E}$ represent the entropy, mutual information, and expectation, respectively. Using Monte Carlo approximation of the posterior, we have

$$\mathcal{H}[p(\boldsymbol{y}|\boldsymbol{x},\mathcal{D})] = \mathcal{H}[\mathbb{E}_{p(\boldsymbol{\theta}|D)}[p(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\theta})]] \tag{3a}$$

$$\approx \mathcal{H}\left[\frac{1}{S}\sum_{s=1}^{S} p(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\theta}^s)\right] \tag{3b}$$

$$\mathbb{E}_{p(\boldsymbol{\theta}|\mathcal{D})}[\mathcal{H}[p(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\theta})]] \approx \frac{1}{S}\sum_{s=1}^{S}\mathcal{H}[p(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\theta}^s)] \tag{3c}$$

where $\boldsymbol{\theta}^s \sim p(\boldsymbol{\theta}|\mathcal{D})$ and $S$ is the number of samples.

### 2.2. Gradient-based Uncertainty Attribution

The gradient-based attribution methods can efficiently generate uncertainty maps via backpropagation. While current CA methods mainly utilize the gradients between the model output and input, some of them can be directly extended for UA by using the gradients from the uncertainty to the input. However, raw gradients can be noisy, necessitating the development of various approaches for smoothing gradients, including Integrated Gradient (IG) [37] with its variants [15, 41], SmoothGrad [34], Grad-cam [31], and FullGrad [36]. Some methods use layer-wise relevance propagation (LRP) to construct classification attributions. Although the LRP-based methods [4, 25, 32] can backpropagate the model outputs layer-wisely to the input, there is no direct extension for the uncertainties since we focus on explaining output variations instead of output values. Moreover, they often require specific NN architectures where

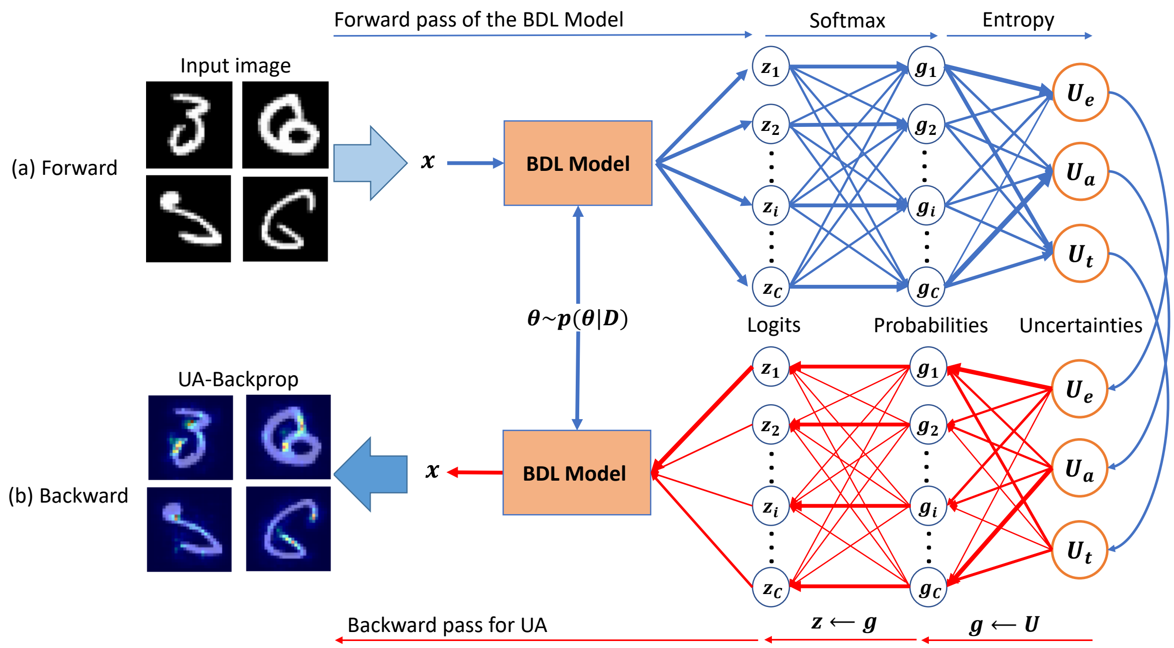


Figure 1. The overall framework of the proposed method. Figure (a) shows the forward propagation of the BDL model for uncertainty quantification. Figure (b) demonstrates the backward process from the uncertainty to the input for attribution analysis, crossing the softmax probabilities, the logits, and the BDL model. The brighter regions indicate higher attributions.

the entropy and softmax functions for uncertainty estimation will violate their requirements. In this paper, we consider the vanilla extension of SmoothGrad, FullGrad, and IG-based methods as baselines. Please refer to Appendix 1 and the survey papers [1, 2, 27] for more discussions.

We contend that the straightforward application of existing attribution methods may not be adequate for conducting UA. Our approach relies on three crucial goals: (1) the uncertainty should be fully attributed with the completeness property satisfied; (2) the pixel-wise attributions should be positive due to data imperfections; (3) the proposed approach should prevent gradient-vanishing issues. Vanilla backpropagation of uncertainty gradients often suffers from the vanishing gradients because of the small magnitude of uncertainty estimates. The resulting visualizations may have "scatter" attributions, which are incomprehensible. Since vanilla adoption of existing methods for deterministic NNs would always violate some of these goals, it is necessary to establish a new gradient-based UA method with competitive accuracy and high efficiency.

## 3. Uncertainty Attribution with UA-Backprop

### 3.1. Overall Framework

As shown in Figure 1, let $\boldsymbol{z}(\boldsymbol{x}, \boldsymbol{\theta}) \in \mathcal{R}^C$ denote the output of the neural network with input $\boldsymbol{x}$ parameterized by $\boldsymbol{\theta}$, which is the probability logit before the softmax layer. The number of classes is represented by $C$. The probability vector $\boldsymbol{g}(\boldsymbol{x}, \boldsymbol{\theta})$ is generated from $\boldsymbol{z}(\boldsymbol{x}, \boldsymbol{\theta})$ through the softmax function, i.e., $\boldsymbol{g}(\boldsymbol{x}, \boldsymbol{\theta}) = \mathrm{softmax}(\boldsymbol{z}(\boldsymbol{x}, \boldsymbol{\theta}))$, where $g_i(\boldsymbol{x}, \boldsymbol{\theta}) = \frac{\exp(z_i(\boldsymbol{x}, \boldsymbol{\theta}))}{\sum_{j=1}^{C} \exp(z_j(\boldsymbol{x}, \boldsymbol{\theta}))}$. For simplicity, we write $\boldsymbol{z}(\boldsymbol{x}, \boldsymbol{\theta})$ as $\boldsymbol{z}$ and $\boldsymbol{g}(\boldsymbol{x}, \boldsymbol{\theta})$ as $\boldsymbol{g}$. Since the complex posterior distribution $p(\boldsymbol{\theta}|\mathcal{D})$ is often intractable, we use a sample-based approximation. We assume that $\{\boldsymbol{\theta}^s\}_{s=1}^S$ are drawn from $p(\boldsymbol{\theta}|\mathcal{D})$, leading to samples $\{\boldsymbol{z}^s\}_{s=1}^S$ and $\{\boldsymbol{g}^s\}_{s=1}^S$. During forward propagation, $\{\boldsymbol{g}^s\}_{s=1}^S$ is used to calculate the epistemic uncertainty $U_e$, aleatoric uncertainty $U_a$, and total uncertainty $U_t$. Let $U$ represent one of the uncertainties in general. For the backpropagation, the uncertainty traverses $U \rightarrow \boldsymbol{g} \rightarrow \boldsymbol{z} \rightarrow \boldsymbol{x}$. The pseudocode for UA-Backprop is provided in Algorithm 1.

Basically, the contribution of each $g_i$ to $U$, referred to as $U_{g_i}$, is first computed. Since the backward pass of the BDL model contains $S$ paths $\boldsymbol{g}^s \rightarrow \boldsymbol{z}^s \rightarrow \boldsymbol{x}$ for $\boldsymbol{\theta}^s \sim p(\boldsymbol{\theta}|\mathcal{D})$, we then obtain the contribution of each $z_i^s$ to $U$, denoted as $U_{z_i^s}$ by exploring all softmax paths $g_j^s \rightarrow z_i^s$ for $j \in [1, \cdots, C]$. Subsequently, $\boldsymbol{z}_i^s \rightarrow \boldsymbol{x}$ is backpropagated. The UA map $M(\boldsymbol{x})$ is then generated as the pixel-wise contribution to the uncertainty, which aggregates all existing paths $\boldsymbol{z}_i^s \rightarrow \boldsymbol{x}$ with different $s \in [1, \cdots, S]$ and $i \in [1, \cdots, C]$. To fully attribute the uncertainty, the completeness property is enforced on $M(\boldsymbol{x})$, as shown in Sec. 3.5. The backward steps are elaborated in the following sections.

### 3.2. Attribution of Softmax Probabilities

In this section, we calculate the attribution of $\boldsymbol{g}$ to uncertainty $U$. For any $i$, we denote the contribution of $g_i$ to

**Algorithm 1** UA-Backprop + FullGrad

**Input:** A BDL model $\boldsymbol{\theta} \sim p(\boldsymbol{\theta}|\mathcal{D})$ with sample approximation $\{\boldsymbol{\theta}^s\}_{s=1}^S$; Normalization hyperparameter $\tau_1, \tau_2$; The target input $\boldsymbol{x}$ for explanation.
**Ouput:** The uncertainty attribution map $M(\boldsymbol{x})$.
**Step 1 ($U \rightarrow \boldsymbol{g}$):** Compute the attribution of softmax probabilities $\{U_{g_j}\}_{j=1}^C$ based on Eq. (4).
**Step 2 ($\boldsymbol{g} \rightarrow \boldsymbol{z}$):** Based on Eq. (5) and Eq. (8), compute the attribution of each logit $U_{z_i}^s$.
**Step 3 ($\boldsymbol{z} \rightarrow \boldsymbol{x}$):** Generate the uncertainty attribution map with the aggregation from all paths $\boldsymbol{z}_i^s \rightarrow \boldsymbol{x}$ based on Eq. (9) and Eq. (10).

$U_e$, $U_a$, and $U_t$ as $U_{e,g_i}$, $U_{a,g_i}$, and $U_{t,g_i}$, respectively. In general, we denote $U_{g_i}$ as the attribution of $g_i$ to $U$. By utilizing Eq. (3), we can express $U_e$, $U_a$, and $U_t$ in terms of $\{\boldsymbol{g}^s\}_{s=1}^S$, and subsequently decompose them into the sum of individual attributions, as shown in the following equation:

$$U_{t,g_i} = -\left(\frac{1}{S}\sum_{s=1}^S g_i^s\right)\log\left(\frac{1}{S}\sum_{s=1}^S g_i^s\right) \tag{4a}$$

$$U_{a,g_i} = \frac{1}{S}\sum_{s=1}^S -g_i^s \log g_i^s \tag{4b}$$

$$U_{e,g_i} = U_{t,g_i} - U_{a,g_i}, \tag{4c}$$

In general, we can observe that $U_{e,g_i}$, $U_{a,g_i}$, $U_{t,g_i}$ only depend on $g_i$ and are independent of other elements of $\boldsymbol{g}$. Moreover, the uncertainties are completely attributed to the softmax probability layer, i.e., $U_t = \sum_{i=1}^C U_{t,g_i}$, $U_a = \sum_{i=1}^C U_{a,g_i}$, $U_e = \sum_{i=1}^C U_{e,g_i}$. When backpropagating the path $\boldsymbol{g}^s \rightarrow \boldsymbol{z}^s$ to get the attribution of logits, $U_{g_i}$ is shared across samples $\{g_i^s\}_{s=1}^S$.

### 3.3. Attribution of Logits

In this section, we aim to derive $U_{e,z_i^s}$, $U_{a,z_i^s}$, and $U_{t,z_i^s}$ as the contribution of $z_i^s$ to $U_e$, $U_a$, and $U_t$ by investigating the path from $\boldsymbol{g}^s$ to $\boldsymbol{z}^s$. We introduce $c_{g_j^s \rightarrow z_i^s} \in (0, 1)$ as the coefficient that represents the proportion of the uncertainty attribution that $z_i^s$ receives from $g_j^s$. Through collecting all the messages from $\{g_j^s\}_{j=1}^C$, the contribution of $z_i^s$ to $U$, donated as $U_{z_i^s}$, is a weighted combination of the attributions received from the previous layer:

$$U_{z_i^s} = \sum_{j=1}^C c_{g_j^s \rightarrow z_i^s} U_{g_j}. \tag{5}$$

To satisfy the completeness property, it is expected that $U_{g_j}$ is fully propagated into the logit layer as shown in the following equation:

$$U_{g_j} = \sum_{i=1}^C c_{g_j^s \rightarrow z_i^s} U_{g_j}, \tag{6}$$

which is a commonly held assumption in many message-passing mechanisms. Eq. (6) indicates that $\sum_{i=1}^C c_{g_j^s \rightarrow z_i^s} = 1$. In this paper, we apply the softmax gradients to determine $c_{g_j^s \rightarrow z_i^s}$ for the backward step from $\boldsymbol{g}^s$ to $\boldsymbol{z}^s$. Specifically, the gradient of $g_j^s$ to $z_i^s$ is as follows:

$$\frac{\partial g_j^s}{\partial z_i^s} = \begin{cases} g_j^s(1-g_j^s) & \text{if } i = j \\ -g_i^s g_j^s & \text{if } i \neq j \end{cases}. \tag{7}$$

Since $\sum_{k=1}^C g_k^s = 1$ due to the definition of softmax function, it is notable that $|\frac{\partial g_i^s}{\partial z_i^s}| > |\frac{\partial g_j^s}{\partial z_i^s}|$ for $i \neq j$, signifying that $g_i^s$ is the primary source of the attribution for $z_i^s$. We normalize the gradients to the obtain the coefficients using $\phi(\cdot)$, with the aim of circumventing extremely small coefficients and thus addressing the gradient-vanishing problem. In this study, $\phi(\cdot)$ is a softmax function with temperature $\tau_1$, i.e.,

$$\begin{aligned} c_{g_j^s \rightarrow z_i^s} &= \phi_i\left(\left\{\frac{\partial g_j^s}{\partial z_k^s}\right\}_{k=1}^C, \tau_1\right) \\ &= \frac{\exp\left(\frac{\partial g_j^s}{\partial z_i^s}/(g_j^s \cdot \tau_1)\right)}{\sum_{k=1}^C \exp\left(\frac{\partial g_j^s}{\partial z_k^s}/(g_j^s \cdot \tau_1)\right)}, \end{aligned} \tag{8}$$

where $g_j^s \cdot \tau_1$ is employed for avoiding uniform or extremely small coefficients. It is expected that $g_i^s$ provides the major contribution to $z_i^s$ since the denominator of the softmax function in $\boldsymbol{z}^s \rightarrow \boldsymbol{g}^s$ serves only as a normalization term.

### 3.4. Attribution of Input

Given the uncertainty attribution $\{U_{z_i^s}\}_{i=1}^C$, associated with $\{z_i^s\}_{i=1}^C$, the attribution map in the input space is generated by backpropagating through $\boldsymbol{z}^s \rightarrow \boldsymbol{x}$. Since each $z_i^s$ may represent different regions of the input, we individually find the corresponding regions of $\boldsymbol{x}$ that contribute to each $z_i^s$, denoted by $M_i^s(\boldsymbol{x})$. Finally, the uncertainty attribution map $M(\boldsymbol{x})$ is derived by a linear combination of $M_i^s(\boldsymbol{x})$ and $U_{z_i^s}$, i.e.,

$$M(\boldsymbol{x}) = \frac{1}{S}\sum_{s=1}^S\sum_{i=1}^C U_{z_i}^s M_i^s(\boldsymbol{x}). \tag{9}$$

$M(\boldsymbol{x})$ indicates the pixel-wise attributions of $U$, which is a two-dimensional matrix that has the same height and width as $\boldsymbol{x}$. It is worth noting that during exploring the possible paths for aggregation, the noisy gradients may be smoothed. We notice that some existing gradient-based methods can be used for exploring the path $\boldsymbol{z}^s \rightarrow \boldsymbol{x}$. For example, the magnitude of the raw gradient can be employed such that $M_i^s(\boldsymbol{x}) = |\frac{\partial z_i^s}{\partial \boldsymbol{x}}|$. Especially, more advanced gradient-based methods such as SmoothGrad [34], Grad-cam [31], and FullGrad [36] can be applied. Intuitively, our proposed method can be a general framework. For the FullGrad method as an example, it aggregates both the gradient of $z_i^s$ with respect to input ($\frac{\partial z_i^s}{\partial \boldsymbol{x}}$) and the gradient of $z_i^s$ with respect to the bias variable $\boldsymbol{b}_l^s$ in each convolutional or fully-connected layer $l$ (i.e., $\frac{\partial z_i^s}{\partial \boldsymbol{b}_l^s}$) to create $M_i^s(\boldsymbol{x})$, i.e.,

$$M_i^s(\boldsymbol{x}) = \psi\left(\left|\frac{\partial z_i^s}{\partial \boldsymbol{x}} \odot \boldsymbol{x}\right| + \sum_l \left|\frac{\partial z_i^s}{\partial \boldsymbol{b}_l^s} \odot \boldsymbol{b}_l^s\right|, \tau_2\right), \quad (10)$$

where $\odot$ is the element-wise product and $|\cdot|$ returns the absolute value. Since different methods will have different scales of $M_i(\boldsymbol{x})$, we apply a post-processing function $\psi$ for normalizing and rescaling the gradients. The function $\psi$ first averages over the channels of $\left|\frac{\partial z_i^s}{\partial \boldsymbol{x}} \odot \boldsymbol{x}\right| + \sum_l \left|\frac{\partial z_i^s}{\partial \boldsymbol{b}_l^s} \odot \boldsymbol{b}_l^s\right|$ and then applies an element-wise softmax function with temperature $\tau_2$. As a general framework, we can leverage the current development of gradient-based attribution methods for deterministic NNs to smooth the gradients and avoid the gradient-vanishing issue.

### 3.5. Special Properties

Our proposed method satisfies the completeness property, shown in the following equation:

$$U = \sum_{i=1}^{C} U_{g_i} = \sum_{i=1}^{C} U_{z_i^s} = \sum_{(u,v)} M(\boldsymbol{x})[u, v], \quad (11)$$

where $(u, v)$ is the index for the entries of $M(\boldsymbol{x})$. The proof can be found in Appendix 1. Our method can also be used with various sensitivity methods for $\boldsymbol{z} \rightarrow \boldsymbol{x}$ to satisfy different properties such as implementation invariance and linearity, which are detailed in Appendix 1.

## 4. Uncertainty Mitigation

Leveraging the insights gained from uncertainty attribution, uncertainty mitigation is to develop an uncertainty-driven mitigation strategy to enhance model performance. In particular, the uncertainty attribution map $M(\boldsymbol{x})$ can be utilized as an attention mechanism by multiplying the inputs or features with $1 - M(\boldsymbol{x})$. This can help filter out problematic input information and improve prediction robustness. However, this approach also assigns high weights to unessential background pixels, which is undesirable. To address this issue, the attention weight $A(\boldsymbol{x})$ is defined by the element-wise product of $(1-M(\boldsymbol{x}))$ and $M(\boldsymbol{x})$ in order to strengthen more informative areas, as shown as follows:

$$A(\boldsymbol{x}) = (1 - M(\boldsymbol{x})) \odot M(\boldsymbol{x}). \quad (12)$$

It is important to note that the attention mechanism can be implemented either in the input space or in the latent space. In this study, we apply $A(\boldsymbol{x})$ in the latent space, while conducting ablation studies for the input-space attentions in Sec. 5.2.3. Let $\{\boldsymbol{h}_k(\boldsymbol{x})\}_{k=1}^K$ with size $K$ be the 2D feature maps generated by the last convolutional layer. We downsample $A(\boldsymbol{x})$ to match the dimensions of $\boldsymbol{h}_k(\boldsymbol{x})$ and utilize $\{(1+\alpha A(\boldsymbol{x}))\odot \boldsymbol{h}_k(\boldsymbol{x})\}_{k=1}^K$ as inputs to the classifier, where $\alpha$ is a hyperparameter that can be tuned. Through retraining using the masked feature maps, the model gains improved accuracy and robustness by ignoring the unimportant background information and the fallacious regions. The complete process is illustrated in Figure 2.

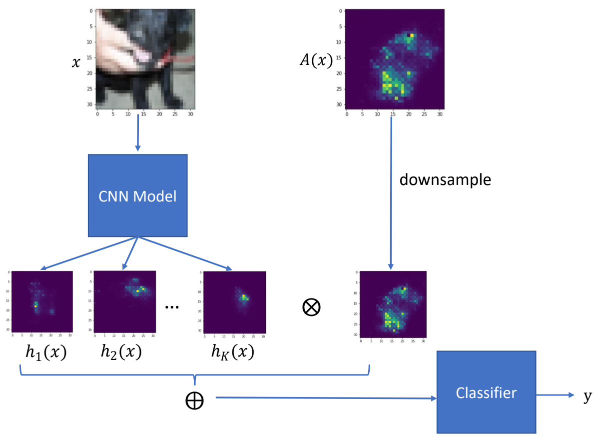


Figure 2. The uncertainty mitigation with attention mechanism.

## 5. Experiments

**Dataset.** We evaluate the proposed method on the benchmark image classification datasets including MNIST [8], SVHN [26], CIFAR-10 (C10) [18], and CIFAR-100 (C100) [17].

**BDL Model.** In our experiments, we use the deep ensemble method [19] for uncertainty quantification, which trains an ensemble of deep neural networks from random initializations. It demonstrates great success in predictive uncertainty calibration and outperforms various approximate Bayesian neural networks [19].

**Implementation Details.** We use standard CNNs for MNIST/SVHN and Resnet18 for C10/C100. The experiment settings, implementation details, and hyperparameters are provided in Appendix 2.

**Baselines.** We compare our proposed method (UA-Backprop + FullGrad) with various baselines on gradient-based uncertainty attribution. The baselines include the vanilla extension of Grad [33], SmoothGrad [34], FullGrad [36], IG [37], and Blur IG [41] for UA. Although CLUE-variants require a generative model and have low efficiency, we include CLUE [3] and $\delta$-CLUE [20] for comparison.

**Evaluation Tasks.** In Sec. 5.1, we qualitatively evaluate the UA performance. In Sec. 5.2, we provide the quantitative

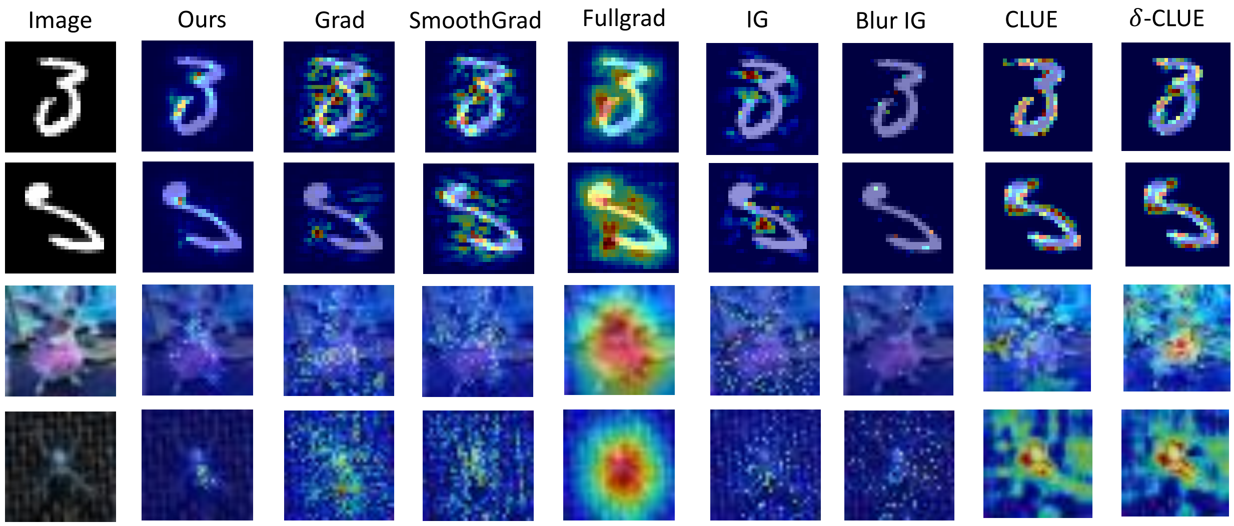


Figure 3. Examples of the epistemic uncertainty attribution maps for various methods on different datasets. Brighter areas indicate essential regions that contribute most to the uncertainty. More examples can be found in Appendix 5.

evaluations including the blurring test, and the attention-based uncertainty mitigation. Various supplementary studies are provided in Appendices 3 and 4.

## 5.1. Qualitative Evaluation

Figure 3 exhibits various examples of attribution maps generated using different techniques. Our analysis reveals that vanilla adoption of CA methods may not be sufficient to generate clear and meaningful visualizations. For instance, as illustrated in Figure 3, we may expect the digit "3" to have a shorter tail, the digit "9" to have a hollow circle with a straight vertical line, and the face of the dog and the small dark body of the spider to be accurately depicted. However, methods such as Grad and Smoothgrad produce ambiguous explanations due to noisy gradients, while FullGrad employs intermediate hidden layers' gradients to identify problematic regions but often lacks detailed information and overemphasizes large central regions. Furthermore, CLUE-based methods tend to identify multiple boundary regions as problematic. They may also fail to provide a comprehensive explanation for complex datasets, where generative models may face significant difficulties in modifying the input to produce an image with lower uncertainty. Finally, CLUE-based methods, Grad, SmoothGrad, and FullGrad fail to fully attribute the uncertainty through the decomposition of pixel-wise contributions. While IG-based methods satisfy the completeness property if the starting image has zero uncertainty, they often produce scattered attributions with minimal regional illustration, posing difficulties in interpretation.

Figure 4 presents various examples of UA maps that depict different types of uncertainties. It is a well-known fact that epistemic uncertainty inversely relates to training data density. Hence, the epistemic uncertainty maps indicate the areas that deviate from the distribution of training data. In some cases, inserting or blurring pixels will help to reduce uncertainty for performance improvement. The aleatoric uncertainty maps quantify the contribution of input noise to prediction uncertainty, which tends to assign high attributions to object boundaries. As displayed in Figure 4, the total uncertainty maps are quite similar to the aleatoric uncertainty maps. That is because the aleatoric uncertainty quantified in Eq. (2) is often much larger than the epistemic uncertainty, which dominates the total uncertainty.

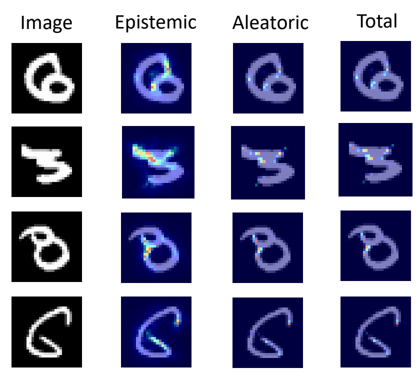


Figure 4. Epistemic, aleatoric, and total uncertainty attribution maps for our proposed method on MNIST dataset.

## 5.2. Quantitative Evaluation

### 5.2.1 Blurring Test

Following [28], we evaluate the proposed method through the blurring test. If the most problematic regions are blurred for a highly uncertain image, we expect a significant uncertainty reduction due to the removal of mislead-

Table 1. Attribution performance in terms of MURR and AUC-URR. We evaluate on four different datasets and blur the image with a maximum of 2% or 5% pixels with the highest contribution to the epistemic uncertainty. The bold values indicate the best performance.

| Method | Maximum Uncertainty Reduction Rate (MURR) ↑ | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | MNIST | | C10 | | C100 | | SVHN | | Avg. Performance |
| | %2 | %5 | %2 | %5 | %2 | %5 | %2 | %5 | %2 + %5 |
| Ours | 0.648 | 0.850 | 0.629 | 0.848 | **0.195** | 0.302 | 0.625 | 0.758 | **0.607** |
| Grad | 0.506 | 0.741 | 0.578 | 0.798 | 0.165 | 0.276 | 0.555 | 0.705 | 0.541 |
| SmoothGrad | 0.601 | 0.779 | 0.566 | 0.800 | 0.154 | 0.255 | 0.575 | 0.735 | 0.558 |
| FullGrad | **0.691** | 0.869 | 0.555 | 0.772 | 0.156 | 0.274 | 0.565 | 0.709 | 0.574 |
| IG | 0.434 | 0.725 | 0.632 | 0.827 | 0.159 | 0.270 | 0.649 | 0.773 | 0.559 |
| Blur IG | 0.305 | 0.515 | **0.693** | **0.971** | 0.184 | **0.318** | **0.762** | **0.896** | 0.581 |
| CLUE | 0.614 | 0.874 | 0.291 | 0.628 | 0.074 | 0.148 | 0.171 | 0.352 | 0.394 |
| $\delta$-CLUE | 0.625 | **0.901** | 0.415 | 0.577 | 0.073 | 0.150 | 0.146 | 0.295 | 0.398 |

| Method | Area under the Uncertainty Reduction Curve (AUC-URR) ↓ | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | MNIST | | C10 | | C100 | | SVHN | | Avg. Performance |
| | %2 | %5 | %2 | %5 | %2 | %5 | %2 | %5 | %2 + %5 |
| Ours | 0.667 | 0.445 | 0.664 | 0.484 | **0.901** | **0.821** | 0.526 | 0.407 | **0.614** |
| Grad | 0.709 | 0.534 | 0.701 | 0.538 | 0.912 | 0.843 | 0.613 | 0.448 | 0.662 |
| SmoothGrad | 0.675 | 0.461 | 0.730 | 0.551 | 0.919 | 0.860 | 0.584 | 0.424 | 0.651 |
| FullGrad | **0.603** | 0.429 | 0.696 | 0.543 | 0.924 | 0.859 | 0.596 | 0.455 | 0.638 |
| Blur IG | 0.816 | 0.667 | **0.638** | 0.466 | 0.914 | 0.851 | 0.541 | 0.402 | 0.662 |
| IG | 0.752 | 0.529 | 0.731 | **0.444** | 0.905 | 0.824 | **0.523** | **0.298** | 0.626 |
| CLUE | 0.709 | 0.397 | 0.861 | 0.624 | 0.966 | 0.926 | 0.919 | 0.815 | 0.777 |
| $\delta$-CLUE | 0.665 | **0.395** | 0.793 | 0.710 | 0.968 | 0.924 | 0.932 | 0.848 | 0.779 |

ing information. The blurring can be conducted via a Gaussian filter with mean 0 and standard derivation $\sigma$. We iteratively blur the pixels based on their contributions to the uncertainty, where we evaluate the corresponding uncertainty reduction curve to demonstrate the effectiveness of our proposed method. Some examples are shown in Figure 5 and the detailed experiment setting is shown in Appendix 2.

The evaluation for the blurring test is conducted on the epistemic uncertainty map since the aleatoric uncertainty captures the input noise and is likely to increase when blurring the image. Denote $v_1, v_2, \cdots, v_T$ as the pixels that

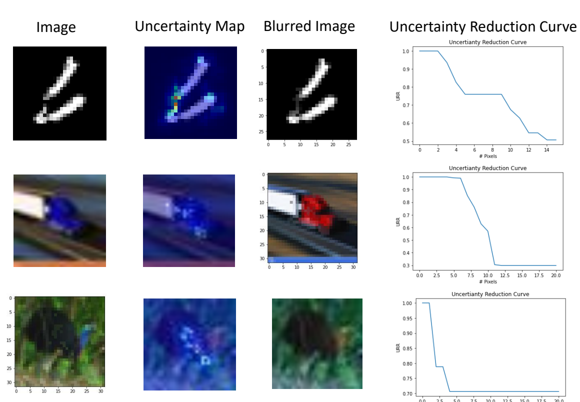


Figure 5. Examples of the blurring test for UA-Backprop.

contribute most to the epistemic uncertainty, following the decreasing order. We iteratively blur up to $t$ pixels, i.e., $\boldsymbol{v}_{1:t}$, and denote the resulting blurred image as $\boldsymbol{x}_t$. The uncertainty reduction rate (URR) shown in Eq. (13) quantifies the extent of achieved uncertainty reduction for blurring up to $t$ problematic pixels:

$$\mathrm{URR}(t) = \frac{1}{|\mathcal{X}|} \sum_{\boldsymbol{x} \in \mathcal{X}} \max_{i \leq t} 1 - \frac{U(\boldsymbol{x}_i)}{U(\boldsymbol{x})}. \tag{13}$$

For URR, we aggregate the results for various sampled images $\boldsymbol{x} \in \mathcal{X}$. The URR curve, obtained by plotting the decreasing normalized values of $\{\mathrm{URR}(t)\}_{t=1}^{T}$, is a key performance metric. We report two evaluation metrics, namely, the maximum uncertainty reduction rate (MURR), i.e., $\max_{t=1:T} \mathrm{URR}(t)$, and the area under the URR curve (AUC-URR). Larger MURR and smaller AUC-URR values indicate superior performance of the UA method. Since the blurring may lead some images to be out-of-distribution, we report median values instead.

As shown in Table 1, our proposed method achieves the best average performance and ranks among the top three in all datasets. In particular, it consistently outperforms Grad, SmoothGrad, FullGrad, and IG. While Blur IG shows promising performance on certain datasets such as C10 and SVHN, it requires a larger number of blurred pixels to achieve improvements and has no advantages to identify the highest problematic regions. Generative-model-based

Table 2. Acc (%) ↑ and NLL ↓ for uncertainty mitigation evaluation. The results are aggregated over 5 independent runs.

| Method | MNIST | | C10 | | C100 | | SVHN | | Avg. Performance | |
|---|---|---|---|---|---|---|---|---|---|---|
| | ACC | NLL | ACC | NLL | ACC | NLL | ACC | NLL | ACC | NLL |
| Ours | 91.95 | **0.287** | **36.48** | **1.768** | 12.12 | 4.326 | **65.13** | **1.489** | **51.42** | **1.968** |
| Grad | 91.35 | 0.302 | 31.60 | 1.938 | 12.13 | 4.422 | 63.74 | 1.578 | 49.71 | 2.060 |
| SmoothGrad | 90.68 | 0.324 | 32.05 | 1.942 | **12.57** | 4.508 | 62.35 | 1.628 | 49.41 | 2.100 |
| FullGrad | 91.39 | 0.300 | 32.85 | 1.920 | 12.06 | 4.574 | 62.38 | 1.568 | 49.67 | 2.091 |
| IG | **91.98** | 0.350 | 34.43 | 1.829 | 11.89 | **4.265** | 64.31 | 1.511 | 50.65 | 1.989 |
| Blur IG | 91.57 | 0.288 | 32.20 | 1.935 | 12.34 | 4.630 | 65.04 | 1.526 | 50.29 | 2.095 |
| CLUE | 91.64 | 0.348 | 33.34 | 1.846 | 12.15 | 4.299 | 60.01 | 1.572 | 49.29 | 2.016 |
| $\delta$-CLUE | 91.76 | 0.350 | 35.02 | 1.809 | 12.22 | 4.362 | 62.71 | 1.612 | 50.43 | 2.033 |
| No attention | 90.78 | 0.358 | 31.62 | 1.921 | 12.02 | 4.536 | 60.64 | 1.569 | 48.77 | 2.096 |

methods, such as CLUE and $\delta$-CLUE, perform well on MNIST but face difficulties in attributing complex images. Additionally, SmoothGrad, Blur IG, and IG require multiple backward passes to attribute one input, while CLUE and $\delta$-CLUE also require a specific optimization process per image, which makes them less efficient. Overall, our proposed method demonstrates superior performance and stands out as the optimal approach for UA in the blurring test.

### 5.2.2 Uncertainty Mitigation Evaluation

Building on the methodology in Sec. 4, we adopt pre-generated attribution maps as attention mechanisms to enhance model performance. The formulation of attention, denoted by $A(\boldsymbol{x})$, is presented in Eq. (12), and is exemplified in Figure 6. To ensure consistency in scale across different methods, the attribution map $M(\boldsymbol{x})$ is normalized using the element-wise softmax function before being used in Eq. (12).

The experimental focus is on training with limited data due to the time-consuming process of generating attribution maps for large datasets, particularly for methods such as Blur IG, SmoothGrad, and CLUE. To this end, we randomly select 500, 1000, 2000, and 4000 images from MNIST, C10, SVHN, and C100, respectively. The selected samples are trained with pre-generated attention maps and evaluated on the original testing data. The evaluation metrics used are accuracy (ACC) and negative log-likelihood (NLL). The experimental setup is detailed in Appendix 2.

Table 2 presents the results obtained for uncertainty mitigation. The method "no attention" refers to plain training without attention incorporated. Our method demonstrates a 6% improvement in ACC compared to vanilla training, suggesting a promising potential for utilizing attribution maps for further model refinement. Our method consistently outperforms other attribution methods in terms of averaged ACC and NLL. We notice that more significant improvement in NLL often occurs for smaller datasets, whereas C100 is challenging to fit with limited samples, and the performance will be more influenced by stochastic training.

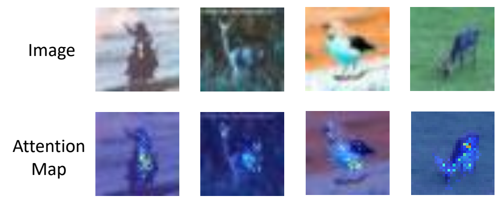

Figure 6. Examples of attention maps for UA-Backprop.

### 5.2.3 Ablation Studies and Further Analysis

To have a comprehensive evaluation, we conduct the anomaly detection experiment in Appendix 3, which compares the predicted problematic regions with the known ground truth. Ablation studies such as efficiency analysis, attribution performances under different experiment settings, and hyperparameter sensitivity analysis are provided in Appendix 4.

## 6. Conclusion

This research aims at developing explainable uncertainty quantification methods for BDL. It will significantly advance the current state of deep learning, allowing it to accurately characterize its uncertainty and improve its performance, facilitating the development of safe, reliable, and trustworthy AI systems. Our proposed method is designed to attribute the uncertainty to the contributions of individual pixels within a single backward pass, resulting in competitive accuracy, relaxed assumptions, and high efficiency. The results of both qualitative and quantitative evaluations suggest that our proposed method has a high potential for producing dependable and comprehensible visualizations and establishing mitigation strategies to reduce uncertainty and improve model performance.

**Acknowledgement:** This work is supported in part by DARPA grant FA8750-17-2-0132 and by the Rensselaer-IBM AI Research Collaboration (http://airc.rpi.edu), part of the IBM AI Horizons Network.

## References


[1] Marco Ancona, Enea Ceolini, Cengiz Öztireli, and Markus Gross. Towards better understanding of gradient-based attribution methods for deep neural networks. *arXiv preprint arXiv:1711.06104*, 2017. 2, 3

[2] Marco Ancona, Enea Ceolini, Cengiz Öztireli, and Markus Gross. Gradient-based attribution methods. In *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*, pages 169–191. Springer, 2019. 3

[3] Javier Antorán, Umang Bhatt, Tameem Adel, Adrian Weller, and José Miguel Hernández-Lobato. Getting a clue: A method for explaining uncertainty estimates. *arXiv preprint arXiv:2006.06848*, 2020. 2, 5

[4] Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PloS one*, 10(7):e0130140, 2015. 2

[5] Eduardo DC Carvalho, Ronald Clark, Andrea Nicastro, and Paul HJ Kelly. Scalable uncertainty for computer vision with functional variational inference. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12003–12013, 2020. 2

[6] 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*, ICML'14, page II–1683–II–1691. JMLR.org, 2014. 2

[7] Piotr Dabkowski and Yarin Gal. Real time image saliency for black box classifiers. *Advances in neural information processing systems*, 30, 2017. 2

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

[9] Stefan Depeweg, Jose-Miguel Hernandez-Lobato, Finale Doshi-Velez, and Steffen Udluft. Decomposition of uncertainty in bayesian deep learning for efficient and risk-sensitive learning. In *International Conference on Machine Learning*, pages 1184–1193. PMLR, 2018. 2

[10] Ruth Fong, Mandela Patrick, and Andrea Vedaldi. Understanding deep networks via extremal perturbations and smooth masks. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 2950–2958, 2019. 2

[11] Ruth C Fong and Andrea Vedaldi. Interpretable explanations of black boxes by meaningful perturbation. In *Proceedings of the IEEE international conference on computer vision*, pages 3429–3437, 2017. 2

[12] 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. 2

[13] W Keith Hastings. Monte carlo sampling methods using markov chains and their applications. 1970. 2

[14] Gao Huang, Yixuan Li, Geoff Pleiss, Zhuang Liu, John E Hopcroft, and Kilian Q Weinberger. Snapshot ensembles: Train 1, get m for free. *arXiv preprint arXiv:1704.00109*, 2017. 2

[15] Andrei Kapishnikov, Subhashini Venugopalan, Besim Avci, Ben Wedin, Michael Terry, and Tolga Bolukbasi. Guided integrated gradients: An adaptive path method for removing noise. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5050–5058, 2021. 2

[16] Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision? *Advances in neural information processing systems*, 30, 2017. 1

[17] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 5

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

[19] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 6402–6413. Curran Associates, Inc., 2017. 2, 5

[20] Dan Ley, Umang Bhatt, and Adrian Weller. {\delta}-clue: Diverse sets of explanations for uncertainty estimates. *arXiv preprint arXiv:2104.06323*, 2021. 2, 5

[21] Dan Ley, Umang Bhatt, and Adrian Weller. Diverse, global and amortised counterfactual explanations for uncertainty estimates. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 7390–7398, 2022. 2

[22] Christos Louizos and Max Welling. Multiplicative normalizing flows for variational Bayesian neural networks. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 2218–2227, International Convention Centre, Sydney, Australia, 06–11 Aug 2017. PMLR. 2

[23] David J. C. MacKay. A Practical Bayesian Framework for Backpropagation Networks. *Neural Computation*, 4(3):448–472, 1992. 2

[24] Wesley J Maddox, Pavel Izmailov, Timur Garipov, Dmitry P Vetrov, and Andrew Gordon Wilson. A simple baseline for bayesian uncertainty in deep learning. In H. Wallach, H. Larochelle, A. Beygelzimer, F. e-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 13132–13143. Curran Associates, Inc., 2019. 2

[25] Grégoire Montavon, Sebastian Lapuschkin, Alexander Binder, Wojciech Samek, and Klaus-Robert Müller. Explaining nonlinear classification decisions with deep taylor decomposition. *Pattern recognition*, 65:211–222, 2017. 2

[26] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. 2011. 5

[27] Ian E Nielsen, Dimah Dera, Ghulam Rasool, Ravi P Ramachandran, and Nidhal Carla Bouaynaya. Robust explainability: A tutorial on gradient-based attribution methods for

deep neural networks. *IEEE Signal Processing Magazine*, 39(4):73–84, 2022. 3

[28] Iker Perez, Piotr Skalski, Alec Barns-Graham, Jason Wong, and David Sutton. Attribution of predictive uncertainties in classification models. In *The 38th Conference on Uncertainty in Artificial Intelligence*, 2022. 2, 6

[29] Vitali Petsiuk, Abir Das, and Kate Saenko. Rise: Randomized input sampling for explanation of black-box models. *arXiv preprint arXiv:1806.07421*, 2018. 2

[30] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. " why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144, 2016. 2

[31] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017. 2, 4

[32] Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. Learning important features through propagating activation differences. In *International conference on machine learning*, pages 3145–3153. PMLR, 2017. 2

[33] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2013. 2, 5

[34] Daniel Smilkov, Nikhil Thorat, Been Kim, Fernanda Viégas, and Martin Wattenberg. Smoothgrad: removing noise by adding noise. *arXiv preprint arXiv:1706.03825*, 2017. 2, 4, 5

[35] Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, and Martin Riedmiller. Striving for simplicity: The all convolutional net. *arXiv preprint arXiv:1412.6806*, 2014. 2

[36] Suraj Srinivas and François Fleuret. Full-gradient representation for neural network visualization. *Advances in neural information processing systems*, 32, 2019. 2, 5

[37] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *International conference on machine learning*, pages 3319–3328. PMLR, 2017. 2, 5

[38] Matias Valdenegro-Toro. Deep sub-ensembles for fast uncertainty estimation in image classification. *arXiv preprint arXiv:1910.08168*, 2019. 2

[39] Yeming Wen, Dustin Tran, and Jimmy Ba. Batchensemble: an alternative approach to efficient ensemble and lifelong learning. *arXiv preprint arXiv:2002.06715*, 2020. 2

[40] Florian Wenzel, Jasper Snoek, Dustin Tran, and Rodolphe Jenatton. Hyperparameter ensembles for robustness and uncertainty quantification. *arXiv preprint arXiv:2006.13570*, 2020. 2

[41] Shawn Xu, Subhashini Venugopalan, and Mukund Sundararajan. Attribution in scale and space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9680–9689, 2020. 2, 5

[42] Qing Yang, Xia Zhu, Jong-Kae Fwu, Yun Ye, Ganmei You, and Yuan Zhu. Mfpp: Morphological fragmental perturbation pyramid for black-box model explanations. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 1376–1383. IEEE, 2021. 2

[43] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pages 818–833. Springer, 2014. 2