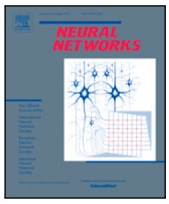
Neural Networks 185 (2025) 107198

Contents lists available at ScienceDirect

Neural Networks

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



Full Length Article

CreINNs: Credal-Set Interval Neural Networks for Uncertainty Estimation in Classification Tasks

Kaizheng Wang [a,e,*], Keivan Shariatmadar [b,e], Shireen Kudukkil Manchingal [c], Fabio Cuzzolin [c], David Moens [d,e], Hans Hallez [a]

[a] DistriNet, Department of Computer Science, Campus Bruges, KU Leuven, Bruges, 8200, Belgium
[b] LMSD, Department of Mechanical Engineering, Campus Bruges, KU Leuven, Bruges, 8200, Belgium
[c] Visual Artificial Intelligence Laboratory, Oxford Brookes University, Oxford, OX3 0BP, UK
[d] LMSD, Department of Mechanical Engineering, Campus De Nayer, KU Leuven, Sint-Katelijne-Waver, 2860, Belgium
[e] Flanders Make@KU Leuven, Leuven, Belgium

ARTICLE INFO

Dataset link: https://github.com/WangKaizheng/CreINNs

Keywords:
Credal sets
Classification
Probability intervals
Uncertainty estimation
Interval neural networks

ABSTRACT

Effective uncertainty estimation is becoming increasingly attractive for enhancing the reliability of neural networks. This work presents a novel approach, termed Credal-Set Interval Neural Networks (CreINNs), for classification. CreINNs retain the fundamental structure of traditional Interval Neural Networks, capturing weight uncertainty through deterministic intervals. CreINNs are designed to predict an upper and a lower probability bound for each class, rather than a single probability value. The probability intervals can define a credal set, facilitating estimating different types of uncertainties associated with predictions. Experiments on standard multiclass and binary classification tasks demonstrate that the proposed CreINNs can achieve superior or comparable quality of uncertainty estimation compared to variational Bayesian Neural Networks (BNNs) and Deep Ensembles. Furthermore, CreINNs significantly reduce the computational complexity of variational BNNs during inference. Moreover, the effective uncertainty quantification of CreINNs is also verified when the input data are intervals.

1. Introduction

Uncertainty-aware neural networks have recently attracted growing interest, as effectively representing and estimating the uncertainties can significantly enhance the reliability and robustness of machine learning systems (Sale, Caprio, & Hüllermeier, 2023), particularly for high-risk and safety-critical applications such as autonomous driving (Fort & Jastrzebski, 2019) and medical sciences (Lambrou, Papadopoulos, & Gammerman, 2010).

Two distinct types of uncertainties, namely *aleatoric uncertainty* (AU) and *epistemic uncertainty* (EU) are widely discussed (Abdar et al., 2021; Hüllermeier & Waegeman, 2021). The former mainly arises from the inherent randomness present in the data generation process and is irreducible, the latter is reducible and caused by the lack of knowledge about the ground-truth network models. Studies (Abdar et al., 2021; Hüllermeier & Waegeman, 2021) indicate that modeling the parameter (weight and bias) uncertainty can contribute to a better estimate of the uncertainty and facilitate reliable inference. The primary justification is that effectively representing parameter uncertainty can yield a collection of plausible network models (Hüllermeier & Waegeman, 2021). These models have the potential to encompass the fundamental network model. As a result, viable second-order uncertainty frameworks can be applied to model the AU and EU in the process and express uncertainty about a prediction's uncertainty (Hüllermeier & Waegeman, 2021; Sale et al., 2023).

In general, uncertainty representation and quantification can be achieved using probabilistic models such as distributions or deterministic methods such as intervals. Compared to probabilistic approaches, intervals usually require fewer assumptions on probability theories and allow for theoretical guarantees on the reliability and robustness of the results (Oala et al., 2021; Sadeghi, De Angelis, & Patelli, 2019). Another significant benefit is that interval models enable handling the interval data (Kowalski & Kulczycki, 2017; Sadeghi et al., 2019; Tretiak, Schollmeyer, & Ferson, 2023). Consequently, applying intervals for uncertainty estimation in neural networks has stimulated considerable research interest and effort. Garczarczyk has introduced *Interval Neural Networks* (INNs) to approximate continuous interval-valued functions, in which their weights and predictions are in the

* Corresponding author.
E-mail addresses: kaizheng.wang@kuleuven.be (K. Wang), keivan.shariatmadar@kuleuven.be (K. Shariatmadar), 19185895@brookes.ac.uk (S.K. Manchingal), fabio.cuzzolin@brookes.ac.uk (F. Cuzzolin), david.moens@kuleuven.be (D. Moens), hans.hallze@kuleuven.be (H. Hallez).

https://doi.org/10.1016/j.neunet.2025.107198
Received 21 August 2024; Received in revised form 17 December 2024; Accepted 19 January 2025
Available online 27 January 2025

form of deterministic intervals (Garczarczyk, 2000). The method was validated by numerical simulation in regression tasks. A subsequent study (Kowalski & Kulczycki, 2017) has extended probabilistic neural networks by incorporating intervals for robust classification. Nevertheless, this approach is specifically designed for inputs represented as interval data and is validated through numerical testing only. Furthermore, the method does not incorporate uncertainty regarding network parameters, thereby excluding the EU entirely. In addition, Sadeghi et al. (2019) have explored efficient training of INNs for imprecise training data in the context of regression tasks. More recently, an INN-based framework has been proposed to produce uncertainty scores and detect failure modes in image reconstruction (Oala et al., 2021). During the training process, an empirical regression-based loss function is deployed to ensure that the resulting real-number prediction intervals contain labels with some probabilities while limiting the ranges of intervals. Tretiak et al. (2023) have investigated the application of original deterministic INNs for imprecise regression (Cattaneo & Wiencierz, 2012) with interval-dependent variables.

Although there has been considerable advancement of INNs in the field of regression tasks, there are notable gaps in the current research on INNs for classification.

(i) Existing INNs typically yield deterministic interval predictions, while traditional neural networks are expected to provide a probability vector over classes in classification tasks. Thus, a research question emerges regarding the reasonable design for assigning probabilities to individual classes based on the interval-formed outputs of INNs.

(ii) In standard settings of classification problems, the labels are one-hot encoded, i.e., the probability value is 1 for the true class and 0 else. This prevents INNs from being reasonably and effectively trained using existing strategies. For example, applying existing regression approaches, such as requiring prediction intervals to include the corresponding labels (Oala et al., 2021), can result in that parameter and prediction intervals collapsing to singular pointwise values.

(iii) There is a lack of empirical studies showcasing the application of existing INNs to more extensive and deep network architectures. For instance, more recent work on INNs (Betancourt & Muhanna, 2022; Lai et al., 2022; Tretiak et al., 2023) has been validated on Multi-layer perceptrons (MLPs) with limited layers.

Given the challenges identified in current studies on INNs, intriguing research questions arise: *Can the existing INN framework be effectively extended to facilitate uncertainty quantification in classification tasks and adapted to modern deep neural network architectures? Furthermore, how well does the proposed neural network estimate uncertainty when provided with standard and interval input data?*

In response, we introduce a novel *Credal-Set Interval Neural Network* (CreINN) for the estimation of uncertainty in classification tasks. CreINNs maintain the fundamental structure of conventional INNs, expressing parameter uncertainty through deterministic intervals. In contrast to the generation of deterministic intervals by conventional INNs or a single probability vector by standard neural networks, CreINNs predict a set of probability intervals (De Campos, Huete, & Moral, 1994) over classes, representing the lower and upper probability bounds across the set of classes. These probability intervals encode a credal set, a convex set of probability distributions (Levi, 1980), for uncertainty quantification. The main novelty and contributions are summarized as follows:

(i) We design an innovative activation function, *Interval SoftMax*, to transform the interval-formed outputs of classical INNs to convex probability intervals that formulate credal set predictions for estimating the aleatoric and epistemic uncertainty.

(ii) We present the strategy of making a unique class index from the outputted probability intervals of CreINNs, based on the so-called *intersection probability transform* (Cuzzolin, 2009, 2022). A new training procedure to enable CreINNs to be trained effectively is also presented.

(iii) We propose *Interval Batch Normalization* based on traditional batch normalization (Ioffe & Szegedy, 2015) to facilitate the adaptability of CreINNs to large and deep network architectures, such as ResNet50 (He, Zhang, Ren, & Sun, 2016).

(vi) We examine the ensemble strategy for CreINNs, inspired by the ensemble of classical INNs for regression tasks (Lai et al., 2022; Pearce, Brintrup, Zaki, & Neely, 2018), aiming to mitigate the effect of network parameter initialization during training and enhance the uncertainty estimation quality.

Experimental validations are conducted in two aspects. **(i)** The standard multiclass classification task involves an out-of-distribution (OOD) detection benchmark (CIFAR10 vs. SVHN dataset) and the binary classification task uses the Chest X-ray dataset. The results demonstrate that CreINN and the ensemble of CreINNs achieve superior or comparable uncertainty quantification compared to probabilistic approaches, such as variational Bayesian Neural Networks (BNNs) (Molchanov, Ashukha, & Vetrov, 2017; Wen, Vicol, Ba, Tran, & Grosse, 2018), Deep Ensembles (Lakshminarayanan, Pritzel, & Blundell, 2017), and the ensemble of BNNs. In addition, the CreINN significantly reduces the computational burden for inference compared to BNNs. **(ii)** Multiclass and binary classification tasks utilizing interval-formed CIFAR10 and X-ray datasets: The effectiveness of uncertainty estimation of CreINNs in interval input cases is verified quantitatively and qualitatively.

The remainder of this paper is organized as follows. Section 2 introduces the background and related work. Section 3 presents our CreINN methodology in full detail. Section 4 describes the experimental validations. Section 5 outlines the conclusions and future work. Appendix A provides the relevant mathematical discussions.

2. Background and related work

This section introduces the concepts of aleatoric and epistemic uncertainty in Section 2.1, as well as probabilistic approaches and interval methods for uncertainty estimation in Sections 2.2 and 2.3, respectively.

2.1. Aleatoric vs. epistemic uncertainty

In supervised learning, a neural network is trained by using a set of independent and identically distributed training data points $\mathbb{D} = \{\boldsymbol{x}_n, \boldsymbol{y}_n\}_{n=1}^{N} \subset \mathbb{X} \times \mathbb{Y}$, where $\mathbb{X}$ and $\mathbb{Y}$ represent the instance and the target space, respectively. In classification tasks involving C elements, the target space $\mathbb{Y}$ consists of a finite collection of class labels, denoted $\mathbb{Y} = \{\text{class}_1, \ldots, \text{class}_k, \ldots, \text{class}_C\}$. Here, $\boldsymbol{y}$ denotes the associated single probability vector. For example, y_k represents the probability value assigned to the kth element class_k.

As the dependence between the input space $\mathbb{X}$ and the target space $\mathbb{Y}$ is not deterministic, neural networks are generally designed to map $\boldsymbol{x}$ to probability distributions on outcomes (Hüllermeier & Waegeman, 2021) to represent the uncertainty of prediction. Standard neural networks (SNNs) typically predict a single probability distribution as the outcome:

$$\boldsymbol{q} = (q_1, \ldots, q_k, \ldots, q_C) \in \mathbb{P}(\mathbb{Y}), \tag{1}$$

where q_k is the predicted probability of kth class instance and $\mathbb{P}(\mathbb{Y})$ denotes the set of all probability measures on the target space $\mathbb{Y}$. SNNs cannot account for EU, as the outputted single probability distribution models the inherent unpredictability between predictions and inputs without considering the uncertainty of how well the predicted distribution approximates the exact dependency (Hüllermeier, Destercke, & Shaker, 2022; Sale et al., 2023). In other words, the pointwise estimates of SNN weights and biases imply full certainty about the ground-truth model.

To fully capture AU and EU, a neural network is desired to implement a mapping of the form $\mathbb{X} \longrightarrow [\![\mathbb{P}(\mathbb{Y})]\!]$, where $[\![\mathbb{P}(\mathbb{Y})]\!]$ represents a second-order framework to express uncertainty about uncertainty (Hüllermeier & Waegeman, 2021; Sale et al., 2023). Among the applicable representation frameworks, Bayesian Neural Networks (BNNs), Deep Ensembles (DEs), and credal-set-based methods incorporate well-established approaches to estimate and differentiate uncertainties associated with predictions.

2.2. Probabilistic uncertainty estimation methods

A dominant probabilistic methodology to estimate and distinguish prediction uncertainty uses BNNs. In BNNs, network weights and biases are modeled as probability distributions. Consequently, the prediction is represented as a second-order distribution, i.e., the probability distribution of distributions (Hüllermeier & Waegeman, 2021). Although suitable approximation techniques, including sampling methods (Hoffman, Gelman, et al., 2014; Neal et al., 2011) and variational inference approaches (Blundell, Cornebise, Kavukcuoglu, & Wierstra, 2015; Gal & Ghahramani, 2016), have been developed for training, and applying Bayesian model averaging (BMA) for inference (Gal & Ghahramani, 2016), the high computational demands of BNNs for training and inference continue to hinder their widespread adoption in practice, particularly in real-time applications (Abdar et al., 2021).

Another important class of methods to effectively quantify prediction uncertainty in a straightforward and scalable manner is Deep Ensembles (Lakshminarayanan et al., 2017). The common way to construct DEs is to aggregate multiple independently trained deterministic neural networks (DNNs), which feature pointwise estimates of network parameters (weights and biases). Recently, DEs have been serving as an established standard to estimate prediction uncertainty (Abe, Buchanan, Pleiss, Zemel, & Cunningham, 2022; Gustafsson, Danelljan, & Schon, 2020; Ovadia et al., 2019). However, DEs are not immune to criticisms, including the lack of robust theoretical foundations and the significant demand for substantial memory complexity, among others (He, Lakshminarayanan, & Teh, 2020; Liu et al., 2020).

An alternative promising representation framework is based on credal sets, a convex set of probability distributions (Corani, Antonucci, & Zaffalon, 2012; Hüllermeier & Waegeman, 2021; Levi, 1980; Sale et al., 2023). Scholars have conducted extensive research to elucidate the utility of credal sets for uncertainty quantification within the broader domain of machine learning, such as Corani et al. (2012), Corani and Zaffalon (2008), Hüllermeier et al. (2022) and Zaffalon (2002). Recently, Caprio et al. (2024) have introduced imprecise BNNs, which model network weights and predictions as credal sets. Although imprecise BNNs exhibit robustness in Bayesian sensitivity analysis, their computational complexity is comparable to that of an ensemble of BNNs, which poses huge challenges for their widespread application.

2.3. Interval uncertainty estimation methods

Research on the use of deterministic intervals in neural networks to represent and quantify uncertainty, known as interval neural networks (INNs), focuses primarily on regression tasks. One line of INN research (Khosravi, Nahavandi, Creighton, & Atiya, 2011; Lai et al., 2022; Pearce et al., 2018; S. Salem, Langseth, & Ramampiaro, 2020) emphasizes generating deterministic interval predictions while keeping the network weights and biases fixed at point estimates. To account for epistemic uncertainty, some researchers, e.g., (Lai et al., 2022; Pearce et al., 2018; S. Salem et al., 2020) have proposed using ensembles of INNs. For example, the variances of the upper and lower prediction bounds across ensemble INN members can be calculated to quantify the EU associated with each prediction bound (Pearce et al., 2018). Unlike traditional INNs that only represent predictions as intervals, an alternative approach models both network weights and biases as intervals (Betancourt & Muhanna, 2022; Cao, Wang, Wang, Xu, & Wang, 2024; Garczarczyk, 2000; Ishibuchi, Tanaka, & Okada, 1993; Oala et al., 2021; Tretiak et al., 2023). This design allows the INN to capture both aleatoric and epistemic uncertainty within an interval framework. A further advantage of these models is their capacity to handle interval-valued input data.

A limited number of studies have explored the use of INNs for classification tasks, primarily due to the practical challenges discussed earlier in the introduction. For example, Kowalski and Kulczycki (2017) extended the probabilistic neural network framework by incorporating interval representations to enhance robustness. This approach is specifically designed to handle interval-valued input data and does not apply to standard machine learning settings using point-valued input data. In addition, the INN is validated only through numerical experiments in a basic network configuration and does not address epistemic uncertainty.

3. Methodology

This section presents our CreINN approach in full detail. As shown in Fig. 1, CreINN retains the traditional INN framework that represents inputs, node outputs, weights, and biases as deterministic intervals, as discussed in Section 3.1. Using the proposed Interval Softmax activation and redundancy reduction, CreINN generates a set of reachable probability intervals (De Campos et al., 1994) over classes from the deterministic interval vector, as detailed in Section 3.2. These reachable probability intervals represent the lower and upper bounds of the probabilities associated with each class, thereby defining a credal set, i.e., a convex set of probability distributions (Levi, 1980). This credal set prediction forms the basis for uncertainty estimation, as explained in Section 3.3. In addition, an intersection probability (Cuzzolin, 2009) can be derived from the probability interval system to make the class prediction, as outlined in Section 3.4. The CreINN training procedure minimizes cross-entropy loss between the label and the intersection probability prediction while ensuring valid weight and bias intervals, as described in Section 3.5. Finally, the proposed Interval Batch Normalization, which supports adaptability to deep network architectures, along with the ensemble strategy for CreINN, are discussed in Sections 3.6 and 3.7, respectively. To enhance clarity, Table 1 provides a list of abbreviations frequently used throughout this work.

3.1. Existing INN structure and CreINN implementation

This section begins with an overview of the existing INN structure in Section 3.1.1, followed by an introduction to the CreINN implementation in Section 3.1.2.

3.1.1. Existing INN structure

Conventional INN employs deterministic interval-formed inputs, outputs, weights, and biases for each node. Forward propagation in its lth layer can be expressed as follows:

$$[\underline{\boldsymbol{a}}, \overline{\boldsymbol{a}}]^l = g^l\left([\underline{\boldsymbol{W}}, \overline{\boldsymbol{W}}]^l \odot [\underline{\boldsymbol{a}}, \overline{\boldsymbol{a}}]^{l-1} \oplus [\underline{\boldsymbol{b}}, \overline{\boldsymbol{b}}]^l\right), \tag{2}$$

where $\oplus$, $\ominus$, and $\odot$ represent interval addition, subtraction, and multiplication, respectively (Hickey, Ju, & Van Emden, 2001). The quantities $[\underline{\boldsymbol{a}}, \overline{\boldsymbol{a}}]^l$, $[\underline{\boldsymbol{a}}, \overline{\boldsymbol{a}}]^{l-1}$, $[\underline{\boldsymbol{W}}, \overline{\boldsymbol{W}}]^l$ and $[\underline{\boldsymbol{b}}, \overline{\boldsymbol{b}}]^l$ are the interval-formed outputs of the lth and the previous $(l-1)$th layer, the weight intervals and bias intervals of the lth layer, respectively. $g^l(\cdot)$ denotes the activation function of the lth layer that is required to be monotonically increasing. Given the standard training set $\mathbb{D} = \{\boldsymbol{x}_n, \boldsymbol{y}_n\}_{n=1}^N$ in machine learning, the model input $[\underline{\boldsymbol{a}}, \overline{\boldsymbol{a}}]^0$ can be set as $\underline{\boldsymbol{a}}^0 = \overline{\boldsymbol{a}}^0 = \boldsymbol{x}$.

The interval arithmetic (Hickey et al., 2001) applied in Eq. (2) endows INNs with the property of "set constraint". Specifically, for any point value $\boldsymbol{a}^{l-1} \in [\underline{\boldsymbol{a}}, \overline{\boldsymbol{a}}]^{l-1}$, $\boldsymbol{W}^l \in [\underline{\boldsymbol{W}}, \overline{\boldsymbol{W}}]^l$, and $\boldsymbol{b}^l \in [\underline{\boldsymbol{b}}, \overline{\boldsymbol{b}}]^l$, the following constraint consistently holds, as follows:

$$\boldsymbol{a}^l = g^l\left(\boldsymbol{W}^l \boldsymbol{a}^{l-1} + \boldsymbol{b}^l\right) \in [\underline{\boldsymbol{a}}, \overline{\boldsymbol{a}}]^l. \tag{3}$$

In the case of non-negative $[\underline{\boldsymbol{a}}, \overline{\boldsymbol{a}}]^{l-1}$, for instance, the output of RELU activation, the forward propagation in Eq. (2) can be simplified as follows:

$$\begin{aligned} \underline{\boldsymbol{a}}^l &= g^l\left(\min\{\underline{\boldsymbol{W}}^l, \boldsymbol{0}\}\overline{\boldsymbol{a}}^{l-1} + \max\{\underline{\boldsymbol{W}}^l, \boldsymbol{0}\}\underline{\boldsymbol{a}}^{l-1} + \underline{\boldsymbol{b}}^l\right) \\ \overline{\boldsymbol{a}}^l &= g^l\left(\max\{\overline{\boldsymbol{W}}^l, \boldsymbol{0}\}\overline{\boldsymbol{a}}^{l-1} + \min\{\overline{\boldsymbol{W}}^l, \boldsymbol{0}\}\underline{\boldsymbol{a}}^{l-1} + \overline{\boldsymbol{b}}^l\right) \end{aligned}. \tag{4}$$

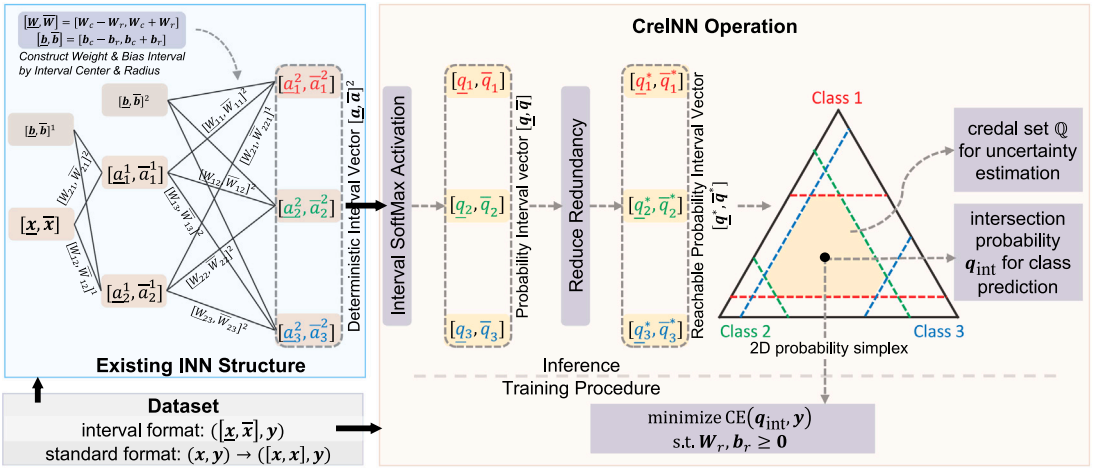


Fig. 1. Illustration of the proposed CreINN model for a three-class classification task. CreINN follows the conventional INN architecture, representing inputs $[\underline{x}, \overline{x}]$, node outputs, weights, and biases (i.e., $[\underline{a}_i^l, \overline{a}_i^l]$ and $[\underline{w}_{ji}^l, \overline{w}_{ji}^l]$ for the ith node of lth layer and $[\underline{b}, \overline{b}]^l$ for the lth layer, respectively) as deterministic intervals. Using the proposed Interval SoftMax activation, a set of probability intervals $[\underline{q}, \overline{q}] := \{[\underline{q}_k, \overline{q}_k]\}_{k=1}^{k=3}$ can derived from the outputted deterministic output interval vector. Through redundancy reduction, the resulting reachable probability interval $[\underline{q}^*, \overline{q}^*]$ (shown as parallel dashed lines) can define a credal set $\mathbb{Q}$ for uncertainty estimation, depicted as the light orange convex hull within the probability simplex (a triangle representing all probability distributions over the target space). In addition, an intersection probability q_{int} can be computed from these probability intervals for class classification purposes. Model training involves minimizing the cross-entropy (CE) loss with constraints that guarantee valid weight and bias intervals. Moreover, the proposed CreINN can handle both interval and standard format data.

Table 1
List of abbreviations.

Abbreviations	Definitions
AU	Aleatoric Uncertainty
AUPRC	Area Under the Precision–Recall Curve
AUROC	Area Under the Receiver Operating Characteristic Curve
BMA	Bayesian Model Averaging
BNNs	Bayesian Neural Networks
BNN-L	Laplace Bridge BNN model
BNN-R	Variational BNN model: Auto-Encoding variational Bayes with the local reparameterization trick
BNN-F	Variational BNN model: Flipout gradient estimator with negative evidence lower bound loss
CE	Cross-entropy Loss
CreINNs	The proposed Credal-Set Interval Neural Networks
DEs	Deep Ensembles
EU	Epistemic Uncertainty
FSVI	Function-space variational inference approach in BNNs
IBN	The proposed Interval Batch Normalization method
ID	In-distribution
INN	Interval Neural Networks
OOD	Out-of-distribution
SNNs	Standard Neural Networks
TU	Total Uncertainty

As the smoothness of Eq. (2) can be guaranteed by some reformulation tricks, as detailed in Appendix A.1, INNs can be trained using standard backward propagation (automatic differentiation) (Oala et al., 2021).

3.1.2. CreINN implementation

In our CreINN, to readily ensure the validity of parameter intervals during propagation, namely $\underline{\boldsymbol{W}} \leq \overline{\boldsymbol{W}}$ and $\underline{\boldsymbol{b}} \leq \overline{\boldsymbol{b}}$, we implement the $[\underline{\boldsymbol{W}}, \overline{\boldsymbol{W}}]$ and $[\underline{\boldsymbol{b}}, \overline{\boldsymbol{b}}]$ in practice by

$$\begin{aligned}[\underline{\boldsymbol{W}}, \overline{\boldsymbol{W}}] &= [\boldsymbol{W}_c - \boldsymbol{W}_r, \boldsymbol{W}_c + \boldsymbol{W}_r] \\ [\underline{\boldsymbol{b}}, \overline{\boldsymbol{b}}] &= [\boldsymbol{b}_c - \boldsymbol{b}_r, \boldsymbol{b}_c + \boldsymbol{b}_r]\end{aligned}, \tag{5}$$

where $\boldsymbol{W}_c$, $\boldsymbol{b}_c$ and $\boldsymbol{W}_r \geq \boldsymbol{0}$, $\boldsymbol{b}_r \geq \boldsymbol{0}$ are the centers (midpoints) and radii (half of ranges) of the weight and bias intervals, respectively. Therefore, the forward propagation in its lth layer of our CreINN is given by Eqs. (2) and (4) with the weight and bias interval as given in Eq. (5).

The CreINN parameter intervals can be efficiently initialized using standard techniques. For example, the centers and radii can be initialized with the default Glorot Uniform initializer (Glorot & Bengio, 2010), applying an additional constraint to ensure non-negative values for the radii. This random initialization, combined with the non-negative constraints on the radii, ensures that most constructed intervals are nonzero when subtracting the center or radius. The empirical results in Fig. 5 further demonstrate the validity of the weight intervals learned after training. To prevent interval explosion during CreINN propagation in deeper neural network architectures, where the upper and lower bounds of node outputs can grow excessively toward infinity, we propose an Interval Batch Normalization (IBN) method, inspired by classical batch normalization. This approach will be discussed in more detail in Section 3.6.

3.2. Credal set prediction generation

For a classification task involving C elements, our CreINNs are designed to transform the outputted interval scores of the final L layer $[\underline{\boldsymbol{a}}, \overline{\boldsymbol{a}}]^L := \{[\underline{a}_k^L, \overline{a}_k^L]\}_k^C$ into a set of probability intervals over C classes, denoted as $[\underline{\boldsymbol{q}}, \overline{\boldsymbol{q}}] := \{[\underline{q}_k, \overline{q}_k]\}_k^C$. The resulting $[\underline{\boldsymbol{q}}, \overline{\boldsymbol{q}}]$ is desired to determine a nonempty credal set, denoted as $\mathbb{Q}$, as follows (De Campos

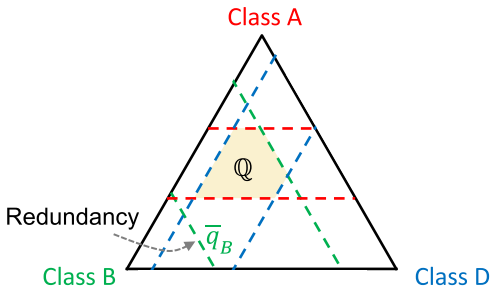


Fig. 2. Redundant probability intervals define a credal set $\mathbb{Q}$ (the convex hull in light orange) in a 2D probability simplex by incorporating interval constraints while some probability bounds (e.g., the upper probability $\overline{q}_B$) may not be reachable.

et al., 1994):

$$\mathbb{Q}=\left\{\boldsymbol{q}\,|\,q_k\in[\underline{q}_k,\overline{q}_k],\ \forall k=1,2,\ldots,C,\ \sum_k^C q_k=1\right\}. \tag{6}$$

The condition guarantees a set of single probability vectors $\boldsymbol{q}$ in $\mathbb{Q}$, whose probability value of each class falls in the corresponding probability interval.

Applying the traditional SoftMax activation function for CreINNs cannot generate valid probability intervals. That is, when computing $[\underline{\boldsymbol{q}},\overline{\boldsymbol{q}}]$ as $\underline{\boldsymbol{q}}=\mathrm{SoftMax}(\underline{\boldsymbol{a}}^L)$ and $\overline{\boldsymbol{q}}=\mathrm{SoftMax}(\overline{\boldsymbol{a}}^L)$, respectively, the resulting probability interval over each class cannot strictly adhere to $\underline{q}_k\le\overline{q}_k$. A numerical example is provided in Appendix A.2.

Inspired by the classical SoftMax, we propose a novel activation, called Interval SoftMax, defined as follows:

$$\begin{aligned}\underline{q}_k&=\frac{\exp(\underline{a}_k^L)}{\exp(\underline{a}_k^L)+\sum_{j\neq k}^C\exp(\frac{\underline{a}_j^L+\overline{a}_j^L}{2})}\\ \overline{q}_k&=\frac{\exp(\overline{a}_k^L)}{\exp(\overline{a}_k^L)+\sum_{j\neq k}^C\exp(\frac{\underline{a}_j^L+\overline{a}_j^L}{2})}\end{aligned}. \tag{7}$$

The original Interval SoftMax holds four useful properties: **(i)** reducing to classical Sigmoid activation function in binary classification; **(ii)** resulting in valid probability intervals and satisfying the constraint in Eq. (6) for defining a nonempty credal set; **(iii)** exhibiting the smoothness for backward propagation; **(iV)** retaining the "set constraint" property described in Eq. (3). Mathematical proofs for four properties of CreINNs are provided in Appendix A.3. The "set constraint" property ensures that the CreINN implicitly and effectively produces a set of standard neural network models which are characterized by weights $\boldsymbol{W}^*\in[\underline{\boldsymbol{W}},\overline{\boldsymbol{W}}]$ and biases $\boldsymbol{b}^*\in[\underline{\boldsymbol{b}},\overline{\boldsymbol{b}}]$. The model predicts a single probability, represented as $\boldsymbol{q}^*$, of which each predicted value q_k^* for the kth class falls within the range $[\underline{q}_k,\overline{q}_k]$.

It should be noted that the probability intervals calculated from the Interval SoftMax may be redundant to determine the credal set resulting from the intersection of all interval constraints, as shown in Fig. 2. Namely, not all upper and lower probability bounds ($\overline{q}_k$ and $\underline{q}_k$ $\forall k$) are guaranteed to be reachable by some probabilities in $\mathbb{Q}$ (De Campos et al., 1994). Here, *reachable* refers to the condition that, for any kth class index of the upper or lower probabilities ($\overline{q}_k$ and $\underline{q}_k$), there at least exists a probability vector $\boldsymbol{q}\in\mathbb{Q}$ such that the kth element of the vector satisfies $q_k=\overline{q}_k$ or $q_k=\underline{q}_k$. Nevertheless, the reachable upper and lower probability bounds of the kth element, represented by $\overline{q}_k^*$ and $\underline{q}_k^*$, respectively, can be readily computed as follows (De Campos et al., 1994):

$$\overline{q}_k^*=\min(\overline{q}_k,1-\sum_{j\neq k}\underline{q}_j),\ \underline{q}_k^*=\max(\underline{q}_k,1-\sum_{j\neq k}\overline{q}_j). \tag{8}$$

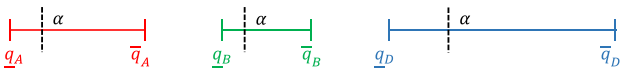


Fig. 3. Determining an intersection probability from the probability interval systems on target space of $\mathbb{Y}=\{A, B, D\}$.

3.3. Uncertainty estimation

The uncertainty quantification for credal sets represents a vibrant area of research (Abellán, Klir, & Moral, 2006; Hüllermeier et al., 2022). Given a credal set $\mathbb{Q}$, a generalization of the Shannon entropy (denoted as H) has been proposed to measure total uncertainty (TU) and aleatoric uncertainty (AU) by calculating the upper and lower Shannon entropy, respectively, as follows (Abellán et al., 2006):

$$\overline{H}(\mathbb{Q})=\underset{\boldsymbol{q}\in\mathbb{Q}}{\text{maximize}}\ H(\boldsymbol{q}),\ \ \underline{H}(\mathbb{Q})=\underset{\boldsymbol{q}\in\mathbb{Q}}{\text{minimize}}\ H(\boldsymbol{q}). \tag{9}$$

The epistemic uncertainty (EU) can be estimated by $\overline{H}(\mathbb{Q})-\underline{H}(\mathbb{Q})$. The calculation of $\overline{H}(\mathbb{Q})$ in CreINNs is by solving the following constrained optimization problem:

$$\begin{aligned}&\overline{H}(\mathbb{Q})=\text{maximize}\sum_k^C-q_k\log_2 q_k\\ &\text{s.t. } q_k\in[\underline{q}_k^*,\overline{q}_k^*]\forall k \text{ and } \sum_k^C q_k=1\end{aligned}, \tag{10}$$

which seeks the highest entropy value of the probability distribution within the credal set. $\underline{H}(\mathbb{Q})$, for which maximize is replaced by minimize, searches for the minimal entropy.

In a special context of binary classification, a single probability interval $[\underline{q},\overline{q}]$ represents the credal set. Recently, more rational and alternative measures have been proposed (Hüllermeier et al., 2022) and applied in our work, as follows:

$$\mathrm{AU}:=\min(\underline{q},1-\overline{q});\ \ \mathrm{EU}:=\overline{q}-\underline{q};\ \ \mathrm{TU}:=\min(1-\underline{q},\overline{q}). \tag{11}$$

For further discussions on the uncertainty measures and their corresponding strengths and weaknesses, we refer to Hüllermeier et al. (2022).

3.4. Class prediction

Predicting classes in the form of probability interval systems (credal sets) is a decision-making problem under uncertainty. To make a unique class prediction, we adopt the intersection probability transform strategy (Cuzzolin, 2009, 2022) to derive a single probability distribution vector $\boldsymbol{q}_{\text{int}}$ from the generated probability intervals. Any kth element of the intersection probability $\boldsymbol{q}_{\text{int}}$ is computed as

$$q_k^*=\underline{q}_k^*+\alpha(\overline{q}_k^*-\underline{q}_k^*), \tag{12}$$

where the unique constant $\alpha\in[0,1]$ can be computed from

$$\alpha=\left(1-\sum_k^C\underline{q}_k^*\right)/\left(\sum_k^C(\overline{q}_k^*-\underline{q}_k^*)\right). \tag{13}$$

Mathematically, the intersection probability formulates a representative of probability interval systems that equally weights the probability interval for each class and satisfies the normalization condition (Cuzzolin, 2009, 2022). An illustration of intersection probability transform in three-component classification is provided in Fig. 3.

As a result, a unique predicted class index can be derived from the intersection probability $\boldsymbol{q}_{\text{int}}$ as $\mathrm{argmax}(\boldsymbol{q}_{\text{int}})$.

3.5. Training procedure

Generally, the cross-entropy (CE) loss is widely utilized for classification. Given a single predicted probability vector $\boldsymbol{q}$ and the corresponding ground truth $\boldsymbol{y}$, the CE measures the Kullback–Leibler divergence between $\boldsymbol{q}$ and $\boldsymbol{y}$ as $\mathrm{CE}(\boldsymbol{q}, \boldsymbol{y}) := -\sum_k^C y_k \log_2 q_k$. However, generalizing the CE to probability interval systems (lower/upper probabilities) is still an open research subject (Lienen, Demir, & Hullermeier, 2023; Song & Deng, 2019; Soubaras, 2011). Considering that the intersection probability represents the most representative single probability for approximating probability interval systems, we employ the intersection probability $\boldsymbol{q}_{\mathrm{int}}$ in CE for CreINN training. Specifically, the training objective is

$$\text{minimize } \frac{1}{N}\sum_n^N \mathrm{CE}(\boldsymbol{q}_{\mathrm{int}_n}, \boldsymbol{y}_n) \quad \text{s.t.} \quad \boldsymbol{W}_r, \boldsymbol{b}_r \geq 0, \tag{14}$$

where N is the number of training samples and the constraint is to ensure the validity of the learned weight and bias intervals during training.

3.6. Interval batch normalization

In modern and deep neural network architectures such as ResNet, batch normalization (Ioffe & Szegedy, 2015) has emerged as an indispensable element. In addition, our tests have also revealed that Interval SoftMax may result in a numerical overflow when the input $[\underline{\boldsymbol{a}}, \overline{\boldsymbol{a}}]^L$ has a wide range.

To enhance the scalability of CreINNs for large and deep architectures, and to mitigate the challenge of numerical overflow, we introduce a novel heuristic approach called Interval Batch Normalization (IBN), derived from the conventional batch normalization methodology. The IBN transform is illustrated in Algorithm 1. Specifically, for mini-batch interval-formed node activations, for instance the outputs of lth layer $[\underline{\boldsymbol{a}}, \overline{\boldsymbol{a}}]^l$, the center and radius (half of ranges) of each interval are computed. The mini-batch centers and radii are then normalized, respectively. Finally, the batch-normalized centers and radii synthesize the normalized deterministic intervals. In addition, the training and inference in batch-normalized CreINNs follow the same procedure as the traditional batch normalization.

Algorithm 1 Interval Batch Normalization Transform

Input: Mini-batch inputs: $\left\{[\underline{a}_i, \overline{a}_i]\right\}_{i=1}^{\eta}$; Hyperparameter ϵ; Trainable parameters γ_c, β_c, γ_r, β_r

Output: $\left\{[\underline{a}_{\mathrm{IBN}_i}, \overline{a}_{\mathrm{IBN}_i}] = \mathrm{IBN}_{\gamma_c,\beta_c,\gamma_r,\beta_r}\left([\underline{a}_i, \overline{a}_i]\right)\right\}_{i=1}^{\eta}$

1. Compute the center and radius of intervals

$\{c_i\} \leftarrow \{\frac{\underline{a}_i + \overline{a}_i}{2}\}$, $\{r_i\} \leftarrow \{\frac{\overline{a}_i - \underline{a}_i}{2}\}$

2. Compute the mini-batch mean and variance

$\mu_{\mathcal{B},c} \leftarrow \frac{1}{\eta}\sum_{i=1}^{\eta} c_i$, $\mu_{\mathcal{B},r} \leftarrow \frac{1}{\eta}\sum_{i=1}^{\eta} r_i$

$\sigma^2_{\mathcal{B},c} \leftarrow \frac{1}{\eta}\sum_{i=1}^{\eta}(c_i - \mu_{B,c})^2$, $\sigma^2_{\mathcal{B},r} \leftarrow \frac{1}{\eta}\sum_{i=1}^{\eta}(r_i - \mu_{B,r})^2$

3. Normalize, scale, and shift

$\hat{c}_i \leftarrow \frac{c_i - \mu_{\mathcal{B},c}}{\sqrt{\sigma^2_{\mathcal{B},c} + \epsilon}}$, $\hat{r}_i \leftarrow \frac{r_i - \mu_{\mathcal{B},r}}{\sqrt{\sigma^2_{\mathcal{B},r} + \epsilon}}$

$c_{out,i} \leftarrow \gamma_c \hat{c}_i + \beta_c$, $r_{out,i} \leftarrow \gamma_r \hat{r}_i + \beta_r$

4. Generate output

$[\underline{a}_{\mathrm{IBN}_i}, \overline{a}_{\mathrm{IBN}_i}] \leftarrow \left[c_{out,i} - |r_{out,i}|, c_{out,i} + |r_{out,i}|\right]$

3.7. Ensemble strategy

To mitigate the influence of different parameter initialization for training and enhance uncertainty estimation performance, we apply a similar ensemble strategy as conventional INNs (Khosravi et al., 2011; Lai et al., 2022; Pearce et al., 2018) for regression to build an ensemble of CreINNs. Specifically, given multiple sets of probability intervals from M distinct CreINNs trained under various parameter initialization settings, we can compute the averaged probability intervals $[\underline{\boldsymbol{q}}_{\mathrm{avg}}, \overline{\boldsymbol{q}}_{\mathrm{avg}}] := \{[\underline{q}_{\mathrm{avg}_k}, \overline{q}_{\mathrm{avg}_k}]\}_k^C$, as follows:

$$\underline{q}_{\mathrm{avg}_k} = \frac{1}{M}\sum_m^M \underline{q}^*_{m_k} \quad \overline{q}_{\mathrm{avg}_k} = \frac{1}{M}\sum_m^M \overline{q}^*_{m_k}, \tag{15}$$

where $[\underline{q}^*_{m_k}, \overline{q}^*_{m_k}]$ represents the reachable probability interval for kth class of the mth ensemble member. It can be proved that the averaged probability intervals can define the nonempty credal set (See Appendix A.4). As a result, the class prediction and uncertainty estimation methods discussed in Section 3.3 are also applicable.

4. Experimental validations

This section describes the experimental validation of CreINNs using the standard datasets in Section 4.1 and interval input data in Section 4.2.

4.1. Classification using standard datasets

In this validation process, we consider multiclass and binary classification problems. The former involves a standard out-of-distribution (OOD) detection benchmark, utilizing CIFAR10 (Krizhevsky, Nair, & Hinton, 2009) as an in-domain and SVHN (Netzer et al., 2011) as an OOD dataset. The latter uses the real-world Chest X-ray dataset (Kermany et al., 2018) in the medical context of pneumonia detection.

4.1.1. Experiment setup

In terms of baselines, we opt for two standardized variational BNNs: **(i)** BNN-R (Auto-Encoding variational Bayes Kingma & Welling, 2013 with the local reparameterization trick Molchanov et al., 2017) and **(ii)** BNN-F (Flipout gradient estimator with negative evidence lower bound loss Wen et al., 2018). BNNs using full sampling approaches are excluded from the comparison due to their extensively higher computational resource requirements (Gawlikowski et al., 2023; Jospin, Laga, Boussaid, Buntine, & Bennamoun, 2022). Moreover, we include the recently proposed tractable functional space variational inference Bayesian model (FSVI) (Rudner, Chen, Teh, & Gal, 2022) and Laplace Bridge BNN (BNN-L) (Hobbhahn, Kristiadi, & Hennig, 2022) as the baselines for comparison. The key distinction between BNN-L and other BNNs lies in: Instead of modeling distributions over the network weights, the BNN-L approximates the full distribution over the softmax outputs of a standard deep network using the Laplace bridge approach, enabling rapid uncertainty estimation (Hobbhahn et al., 2022). Standard neural networks (SNNs) are also considered. All models are implemented on the ResNet50 architecture for the CIFAR10 dataset and the ResNet18 architecture for the X-ray dataset. Furthermore, BNN-R Ensemble, BNN-F Ensemble, FSVI Ensemble, BNN-L Ensemble, CreINN Ensemble, and Deep Ensembles (DEs) are constructed for performance comparison by combining five single models trained with distinct random seeds. Deep Ensembles consist of ten SNNs to retain a nearly equivalent parameter count. Deep Ensembles normally serve as the strong uncertainty baseline (Abe et al., 2022; Gustafsson et al., 2020; Mucsányi, Kirchhof, & Oh, 2024).

Regarding training details, we utilize a single Tesla P100-SXM2-16 GB GPU as the training device. The training and validation data split for the CIFAR10 and the X-ray dataset is the classic 5:1. The Adam optimizer is applied with a learning rate scheduler, initialized at 0.001. The learning rate is subject to a reduction of 0.1 at epochs 80 and 120 for the CIFAR10 dataset and 25 epochs for the X-ray dataset, respectively. Following the recommendations of the original study (Hobbhahn et al., 2022), we train BNN-L using our pre-trained SNN models under the default experimental settings of the BNN-L approach. Standard data augmentation is also uniformly implemented across all models. As the FSVI approach requires highly customized code implementation, we used the official FSVI repository (Rudner

et al., 2022) in our experiments, along with all its training configurations (e.g., a different learning rate scheduler, selecting and saving the best model during training, etc.). Each model is trained over 15 runs for statistical significance.

4.1.2. Uncertainty evaluation metrics

As there is no ground truth for prediction uncertainty and it is infeasible to directly compare the uncertainty values in different representation formats (namely the set of distributions for DEs and BNNs applying BMA, and probability-interval-based credal sets of CreINNs), we employ two indirect methodologies (downstream tasks) to evaluate the uncertainty evaluation of CreINNs.

(i) Accuracy-rejection (AR) curves for in-distribution (ID) samples. AR curves illustrate the accuracy of a model's prediction as a function of the rejection rate in selective classification (Hühn & Hüllermeier, 2008; Hüllermeier et al., 2022). When processing a batch of instances, those with higher uncertainty are rejected initially, and then the accuracy of the remaining test samples is calculated.

In this work, we separately use the model's aleatoric uncertainty (AU), epistemic uncertainty (EU), and total uncertainty (TU) estimate to reject ID samples. AR curve exhibits a monotonic increase when prediction uncertainty estimation is valid. Conversely, the curve demonstrates a flat profile when random abstention is employed (Hüllermeier et al., 2022). In addition, the area under the AR curve (AUARC) is also used as a measure for comparison (Jaeger, Lüth, Klein, & Bungert, 2023). A higher AUARC score indicates superior performance.

(ii) OOD detection. As the uncertainty-aware models are expected to exhibit greater EU on OOD samples than ID data, a better OOD detection could indicate a higher quality of EU quantification (Hendrycks & Gimpel, 2017; Mukhoti, Kirsch, van Amersfoort, Torr, & Gal, 2023). In addition, we also evaluate the TU estimation in this setting, as TU is a widely used uncertainty measure within BNNs and DEs.

We label ID and OOD samples as zeros and ones in the OOD detection process, respectively. The OOD detection is treated as a binary classification, the uncertainty estimation of the model for each sample is the "prediction". AUROC (Area Under the Receiver Operating Characteristic curve) and AUPRC (Area Under the Precision–Recall curve) scores are used as OOD detection metrics. AUROC quantifies the rates of true and false positives, whereas AUPRC evaluates precision and recall trade-offs, providing valuable insights into the model's effectiveness across different confidence levels. Greater scores indicate a higher OOD detection performance. The OOD detection process is summarized in Algorithm 2.

Algorithm 2 OOD Detection Process

Input: Uncertainty estimates for ID and OOD samples, namely $\boldsymbol{u}_{\text{ID}}$, $\boldsymbol{u}_{\text{OOD}}$
Output: AUROC and AUPRC scores
1. Set labels ($\boldsymbol{b}_{\text{ID}}$) as 0 for ID samples
$\boldsymbol{b}_{\text{ID}} \leftarrow$ zeros(shape of $\boldsymbol{u}_{\text{ID}}$)
2. Set labels ($\boldsymbol{b}_{\text{OOD}}$) as 1 for OOD samples
$\boldsymbol{b}_{\text{OOD}} \leftarrow$ ones(shape of $\boldsymbol{u}_{\text{OOD}}$)
3. Concatenate labels for all samples
$\boldsymbol{b} \leftarrow$ concatenate($\boldsymbol{b}_{\text{ID}}, \boldsymbol{b}_{\text{OOD}}$)
4. Concatenate uncertainty estimates as "predictions"
$\boldsymbol{u} \leftarrow$ concatenate($\boldsymbol{u}_{\text{ID}}, \boldsymbol{u}_{\text{OOD}}$)
5. Compute AUROC and AUPRC values
AUROC $\leftarrow$ roc_auc_score($\boldsymbol{b}, \boldsymbol{u}$)
AUPRC $\leftarrow$ average_precision_score($\boldsymbol{b}, \boldsymbol{u}$)

4.1.3. Results and discussions in multiclass case

Fig. 4(a) shows the averaged training and validation accuracy curves of various single models over 15 runs to monitor the training process. In addition, CreINNs are desired to learn reasonable weight intervals, i.e. the radius (half range of intervals) of the parameter

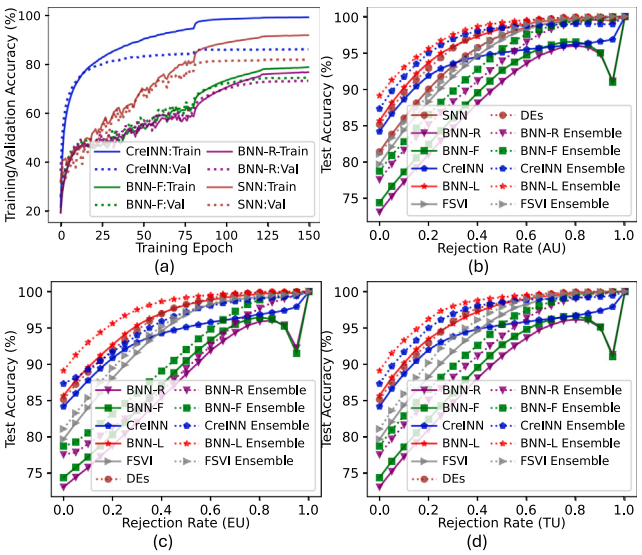


Fig. 4. Monitoring the standard training processes of CreINN, SNN, BNN-F, and BNN-R (a) and AR curves using AU (b), EU (c), and TU (d) estimates, averaged over 15 experimental runs.

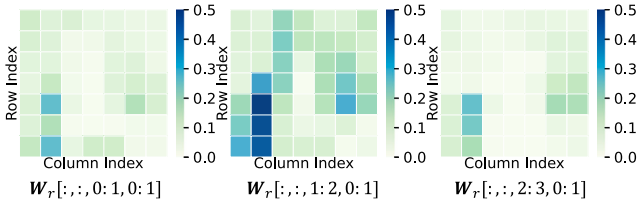


Fig. 5. Heat map of a slice of $\boldsymbol{W}_r$, which has a shape of (7, 7, 3, 64) and is derived from the first convolution layer of a trained CreINN.

intervals in Eq. (5) do not collapse into single values. To verify this, we examine whether $\boldsymbol{W}_r \neq \boldsymbol{0}$ holds after CreINN training. As the weight of the ResNet50 convolutional layers is with a shape of (height, width, channels-in, channels-out) and high-dimensional, we merely demonstrate a slice of $\boldsymbol{W}_r$ in Fig. 5. The heat map verifies that $\boldsymbol{W}_r \neq \boldsymbol{0}$, suggesting that the learned weight intervals do not collapse into single values.

In terms of uncertainty estimation on ID instances, the AR curves in Fig. 4 validate that CreINNs and CreINN Ensemble can effectively estimate the AU, EU, and TU, as evidenced by the positive correlation between accuracy and rejection rate. In contrast, the results of BNNs indicate a notable negative correlation when the rejection rate exceeds approximately 0.85. This observation suggests that BNNs may express higher uncertainty estimates for instances correctly classified within the remaining samples. As illustrated in Table 2 (left), the CreINNs and CreINN ensemble demonstrate the most favorable AUARC compared to other models in an individual or ensemble configuration except BNN-L models. The outperformance appears mainly due to post-training on the well-trained SNN model, the basic test accuracy of BNN-L is largely improved compared to SNNs.

Regarding uncertainty estimation for OOD detection, Table 2 (right) demonstrates the outperformance of CreINNs, evidenced by either the best or second-best AUROC and AUPRC values compared to alternative methods. The enhanced quality of uncertainty estimates is probably mainly beneficial from modeling EU using credal sets rather than distributions. Credal sets integrate sets and distributions within a consistent framework, EU is measured through the assessment of non-specificity across distributions (Hüllermeier & Waegeman, 2021).

Furthermore, Fig. 4 and Table 2 also show that the ensemble strategy can enhance the uncertainty estimation on ID samples and for

Table 2
Performance comparison across various models regarding uncertainty estimation on ID and OOD samples. Note that MBA with samples $N_p = 10$ is applied for inference of single BNN-R, BNN-L, and BNN-L. The best and second-best performances are in **black bold** and **gray bold**, respectively.

		ID Evaluation				OOD Evaluation			
			AUARC			AUROC		AUPRC	
		Test accuracy (%)	AU	EU	TU	EU	TU	EU	TU
Single model	SNN	81.40 ± 1.48	0.950 ± 0.008	–	–	–	–	–	–
	CreINN	84.20 ± 0.30	0.939 ± 0.004	0.939 ± 0.003	0.943 ± 0.002	0.727 ± 0.022	0.745 ± 0.016	**0.854 ± 0.014**	**0.874 ± 0.009**
	BNN-F	74.44 ± 2.44	0.896 ± 0.022	0.884 ± 0.025	0.897 ± 0.021	0.702 ± 0.044	0.738 ± 0.026	0.820 ± 0.030	0.829 ± 0.017
	BNN-R	73.13 ± 3.59	0.886 ± 0.029	0.872 ± 0.033	0.887 ± 0.029	0.703 ± 0.036	0.734 ± 0.020	0.824 ± 0.025	0.827 ± 0.016
	FSVI	79.71 ± 0.53	0.943 ± 0.002	0.941 ± 0.002	0.943 ± 0.002	0.725 ± 0.028	0.711 ± 0.022	0.708 ± 0.036	0.676 ± 0.027
	BNN-L	**85.67 ± 0.33**	**0.967 ± 0.002**	**0.963 ± 0.002**	**0.966 ± 0.002**	**0.745 ± 0.035**	**0.761 ± 0.030**	0.846 ± 0.025	0.859 ± 0.018
Ensemble model	Deep Ensembles	85.16 ± 0.27	0.966 ± 0.001	0.962 ± 0.001	0.966 ± 0.001	0.783 ± 0.006	0.796 ± 0.005	0.873 ± 0.006	0.865 ± 0.004
	CreINN Ensemble	87.32 ± 0.22	0.969 ± 0.001	0.957 ± 0.001	0.970 ± 0.001	0.791 ± 0.010	**0.895 ± 0.003**	0.877 ± 0.008	**0.948 ± 0.002**
	BNN-F Ensemble	78.75 ± 0.90	0.932 ± 0.004	0.911 ± 0.006	0.932 ± 0.004	0.680 ± 0.029	0.758 ± 0.007	0.791 ± 0.026	0.836 ± 0.005
	BNN-R Ensemble	77.58 ± 1.14	0.923 ± 0.008	0.893 ± 0.012	0.922 ± 0.008	0.678 ± 0.018	0.764 ± 0.010	0.802 ± 0.014	0.839 ± 0.004
	FSVI Ensemble	81.11 ± 0.98	0.950 ± 0.004	0.947 ± 0.011	0.953 ± 0.004	**0.845 ± 0.051**	0.821 ± 0.042	0.834 ± 0.051	0.762 ± 0.043
	BNN-L Ensemble	**89.12 ± 0.16**	**0.978 ± 0.000**	**0.978 ± 0.000**	**0.980 ± 0.000**	0.838 ± 0.016	0.834 ± 0.017	**0.904 ± 0.009**	0.896 ± 0.010

Table 3
Relative inference time (multiples) of different models compared to SNN baseline. The MBA with $N_p = 10$ is applied for all BNN models.

SNN	CreINN	BNN-R	BNN-F	BNN-L	FSVI
1.0	4.56	120.15	133.05	1.38	8.60

OOD detection.

In addition to the uncertainty estimation comparison, we report the inference complexity of different models in Table 3. The inference time indicates the time cost of a single instance from the CIFAR10 dataset and is measured by a single Tesla P100 GPU. CreINNs show significant outperformance compared to variational BNNs in inference computational complexity as a single model that can handle EU estimation. This is because CreINNs utilize conventional forward and backward propagation methods and estimate uncertainty during inference without sampling. In contrast, BNNs require costly BMA techniques to capture uncertainties in predictions (10 samplings in Table 3). Although the theoretical scaling of CreINNs is linear in the cost of evaluating the underlying prediction SNNs with a constant factor of 2 (Oala et al., 2021), our experimental results indicate that an inference time overhead of CreINNs is more than four times higher than that of an SNN. The reasons are in two folds. **(i)** The direct implementation of interval arithmetic in Eq. (4) results in instructions four times. **(ii)** The inference cost comparison is less equitable for CreINNs, as they incorporate custom layers without optimization, unlike the other standardized TensorFlow models. Therefore, further optimization at the code implementation level is desired to improve the inference speed and reduce memory consumption for future applications. It is also observed that BNN-L results in a slight increase in inference complexity in comparison to the SNN. This is because BNN-L only approximates the full distribution over the softmax outputs, utilizing the Laplace bridge, while maintaining the architectural structure of the SNN. In addition, due to the highly customized code implementation of FSVI, the comparison of inference complexity in Table 3 may be biased.

As previously stated in Section 3, the distinctive capability of CreINNs compared to alternative probabilistic methodologies is their capacity to process interval input data. We further show that the CreINN model, which is trained on standard input data, is also capable of providing valid uncertainty estimates when presented with internal data, as illustrated in Fig. B.1 in Appendix B.

4.1.4. Results and discussions in binary case

Fig. 6(a) shows the averaged training and validation accuracy curves of the CreINN and SNN over 15 runs to monitor the training process. The results of the BNN-R, BNN-F, and FSVI are excluded, as they are severely underfitting using the X-ray training data in the same training settings. BNN-L is not implemented in this case, as the Laplace bridge is specifically designed for softmax outputs in multiclass

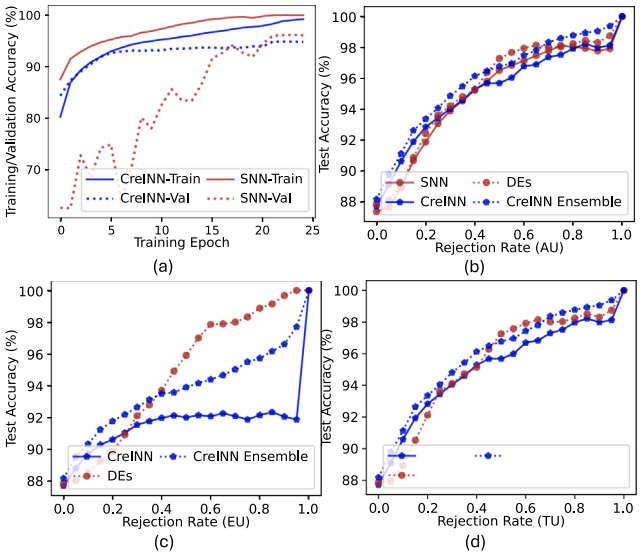


Fig. 6. Training process monitor using the X-ray dataset (a) and AR curves using AU (b), EU (c), and TU (d) estimates, averaged over 15 experimental runs.

cases (Hobbhahn et al., 2022). As a result, the primary objective is to compare the uncertainty estimation performance on ID samples of the CreINN, SNN, CreINN Ensemble, and Deep Ensembles.

Fig. 6 and Table 4 illustrate AR curves and AUARC values on ID samples of various models that employ AU, EU, and TU as rejection metrics, respectively. The positive correlation between accuracy and rejection rate verifies the effectiveness of the uncertainty estimation of CreINNs. In comparison to SNNs and Deep Ensembles, CreINNs, and CreINN Ensemble achieve enhanced or comparable AUARC values. It is also observed that applying the ensemble strategy can enhance the quantification of uncertainties.

Similarly, in binary cases, we also verified that the CreINN model, which is trained on standard input data, can provide valid uncertainty estimates when presented with internal data, as shown in Fig. B.2 in Appendix B.

4.2. Classification using interval input datasets

In this validation process, we consider multiclass and binary classification problems with interval input data constructed from the CIFAR-10 and X-ray datasets.

4.2.1. Data preparation

Despite numerous studies related to interval data, such as (Faza, Shariatmadar, Hallez, & Moens, 2024; Jurio, Pagola, Mesiar, Beliakov,

Table 4
Performance comparison regarding uncertainty estimation on X-ray ID samples.

	Test accuracy (%)	AUARC		
		AU	EU	TU
SNN	87.37 ± 0.59	0.950 ± 0.005	–	–
CreINN	**87.71 ± 0.26**	**0.952 ± 0.005**	**0.916 ± 0.013**	**0.952 ± 0.005**
Deep Ensembles	87.87 ± 0.15	0.955 ± 0.001	**0.948 ± 0.001**	0.954 ± 0.001
CreINN Ensemble	**88.18 ± 0.24**	**0.960 ± 0.002**	0.938 ± 0.007	**0.960 ± 0.002**

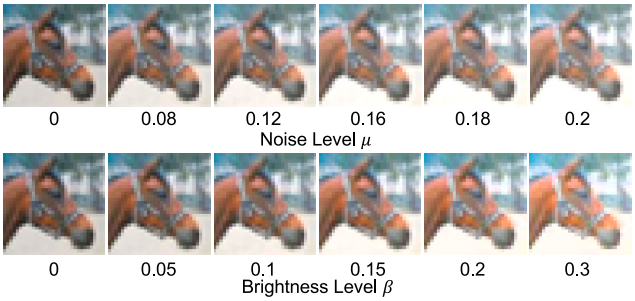


Fig. 7. A generated image sample from the original CIFAR10 dataset using different noise and brightness levels.

& Bustince, 2011) and (Vovan, Phamtoan, Tuan, & Nguyentrang, 2021), there appear to be no open-source interval image datasets in the research community. To investigate the efficacy of CreINNs in addressing challenging interval input data classification tasks, we construct a series of interval image data from the existing standard dataset. The interval image data are used to simulate two different real-world scenarios as follows:

(i) The level of noise or disturbance μ of input measurement falls within a known range. Given an instance $\boldsymbol{x}$ from the original dataset, the μ-level disturbed sample, denoted as $\boldsymbol{x}_\mu$, is generated as follows:

$$\boldsymbol{x}_\mu = \mathrm{Clip}(\boldsymbol{x} + \mu, 0, 1). \tag{16}$$

μ is selected from $\{0, 0.08, 0.12, 0.16, 0.18, 0.2\}$. The function Clip guarantees that $\boldsymbol{x}_\mu$ is a valid representation of an image. More specifically, the input interval for CreINNs $[\underline{\boldsymbol{x}}, \overline{\boldsymbol{x}}] := [\boldsymbol{x}_{\mu=0}, \boldsymbol{x}_{\mu=0.08}]$ implies that images is taken with disturbance level $\mu \in [0, 0.08]$.

(ii) The brightness condition of images β is maintained within a known interval. An RGB image instance with added β-level brightness, represented as $\boldsymbol{x}_\beta$, can be obtained as follows (Hendrycks & Dietterich, 2019):

$$\begin{aligned} \boldsymbol{x}_{\mathrm{hsv}} &= \mathrm{Rgb2Hsv}(\boldsymbol{x}) \\ \boldsymbol{x}_{\mathrm{hsv}}[:,:,2] &= \mathrm{Clip}(\boldsymbol{x}_{\mathrm{hsv}}[:,:,2] + \beta, 0, 1). \\ \boldsymbol{x}_\beta &= \mathrm{Clip}(\mathrm{Hsv2Rgb}(\boldsymbol{x}_{\mathrm{hsv}}), 0, 1) \end{aligned} \tag{17}$$

Here, functions Rgb2Hsv and Hsv2Rgb transform the input from the RGB format to the HSV format and vice versa. β can be chosen from $\{0, 0.05, 0.1, 0.15, 0.2, 0.3\}$. To illustrate, the input interval $[\underline{\boldsymbol{x}}, \overline{\boldsymbol{x}}] := [\boldsymbol{x}_{\beta=0}, \boldsymbol{x}_{\beta=0.05}]$ assumes that the images can be captured with the brightness level $\beta \in [0, 0.05]$.

Fig. 7 illustrates a generated image sample from the original CIFAR10 dataset under different noise and brightness levels.

4.2.2. Experiment setup

Since the baseline models in Section 4.1 are not capable of handling interval input data, we only evaluate CreINNs and CreINN ensemble quantitatively and qualitatively for uncertainty estimation. Regarding the training data, we constructed $[\boldsymbol{x}_{\mu=0}, \boldsymbol{x}_{\mu=0.08}]$ (considering the noise interval) and $[\boldsymbol{x}_{\beta=0}, \boldsymbol{x}_{\beta=0.05}]$ (considering the brightness interval) from the original training set of the CIFAR10 and X-ray datasets. The CreINNs are trained using the same configurations (optimizer, training epochs, etc.) as described in Section 4.1. In terms of test data, we

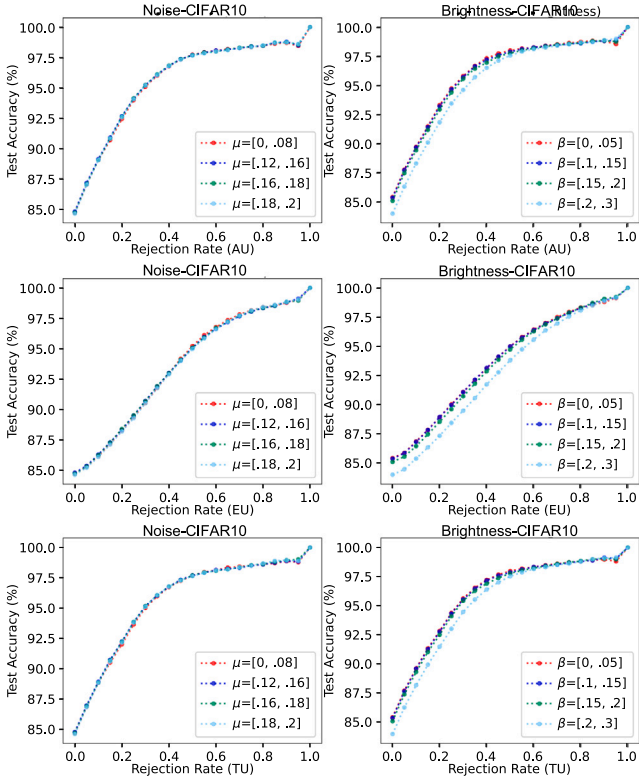


Fig. 8. AR curves using AU, EU, and TU estimates in multiple cases where the input intervals are constructed from CIFAR10 using different noise levels μ and brightness levels β. The results are averaged over 15 experimental runs.

construct different $[\boldsymbol{x}_{\mu_1}, \boldsymbol{x}_{\mu_2}]$ and $[\boldsymbol{x}_{\beta_1}, \boldsymbol{x}_{\beta_2}]$ from the original test set of the CIFAR10 and X-ray datasets. The design intervals of μ and β, $[\mu_1, \mu_2]$ and $[\beta_1, \beta_2]$, were selected as follows:

$$\begin{aligned} [\mu_1, \mu_2] &= [0, .08], [.12, .16], [.16, .18], [.18, .2] \\ [\beta_1, \beta_2] &= [0, .05], [.1, .15], [.15, .2], [.2, .3] \end{aligned} \tag{18}$$

4.2.3. Uncertainty evaluation metrics

In addition to utilizing AR curves as a means of assessing uncertainty on ID samples, as outlined in Section 4.1.2, we also examine the AU, EU, and TU estimation of CreINNs in diverse test interval data, constructed through the use of varying design intervals for noise and brightness (shown in Eq. (18)). To facilitate the examination, we define a measure, called Relative Increase r, as follows:

$$\begin{aligned} r_{[\mu_1,\mu_2]} &:= \frac{1}{E}\sum_{e}^{E}\frac{1}{N_t}\sum_{n_t}^{N_t}\frac{U_{[\mu_1,\mu_2],n_t,e}}{U_{[0,.08],n_t,e}} \\ r_{[\beta_1,\beta_2]} &:= \frac{1}{E}\sum_{e}^{E}\frac{1}{N_t}\sum_{n_t}^{N_t}\frac{U_{[\beta_1,\beta_2],n_t,e}}{U_{[0,.05],n_t,e}} \end{aligned}, \tag{19}$$

where N_t denotes the number of test samples, $E = 15$ represents the number of experimental runs. The notation $U_{[\mu_1,\mu_2],n_t,e}$ represents the AU, EU, or TU estimate of the eth model on the n_t test sample,

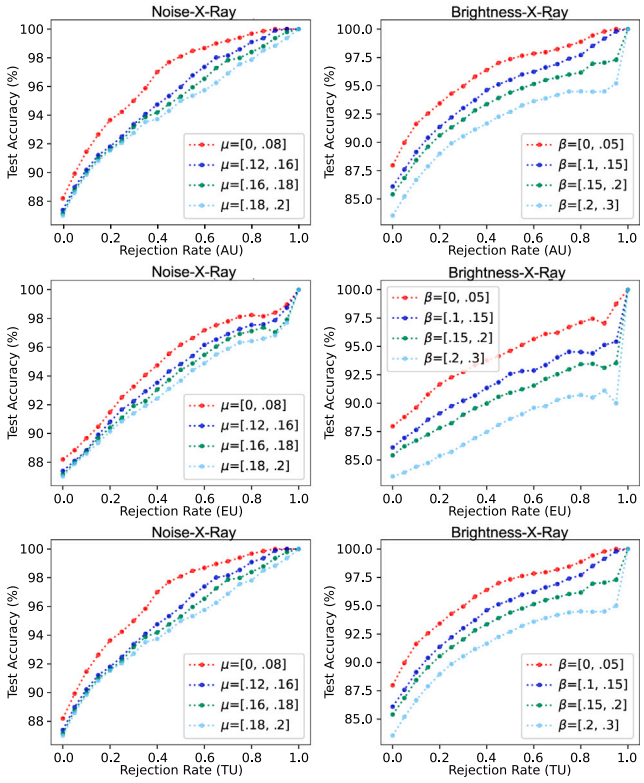


Fig. 9. AR curves using AU, EU, and TU estimates in multiple cases where the input intervals are constructed from X-ray using different noise levels μ and brightness levels β. The results are averaged over 15 experimental runs.

constructed using the noise interval $[\mu_1, \mu_2]$ as defined in Eq. (18). Similarly, the notion $U_{[\beta_1,\beta_2],n_t,e}$ is used for the interval test instance designed by the brightness interval $[\beta_1, \beta_2]$.

4.2.4. Results and discussions

Figs. 8 and 9 demonstrate the AR curves of the CreINN Ensemble in multiple cases where the interval instances are constructed from the CIFAR10 and X-ray datasets using different noise levels μ and brightness levels β. The positive correlation between accuracy and rejection rate in each case verifies the effectiveness of uncertainty quantification. Fig. 10 further demonstrates the relative increase (defined in Eq. (19)) of AU, EU, and TU estimates in interval test instances at different levels of noise μ and brightness β. As the level increases, the uncertainty estimates increase significantly. The evidence verifies CreINNs' capacity to estimate uncertainty in interval input data.

5. Conclusion and future work

In this paper, we introduced innovative CreINNs, which maintain the foundational structure of conventional INNs and can produce credal sets via probability intervals to estimate uncertainty in classification tasks. In addition, the ensemble of CreINNs was also investigated. Experiments using standard image and interval-formed image datasets in multiclass and binary classification tasks verified the proposed methods.

(i) Concerning standard datasets, the CreINN and the ensemble of CreINNs have demonstrated superior or comparable quality of uncertainty quantification compared to variational BNNs, Deep Ensembles, and the ensemble of BNNs. Furthermore, the CreINN markedly reduces the computational burden for inference compared to some variational BNNs.

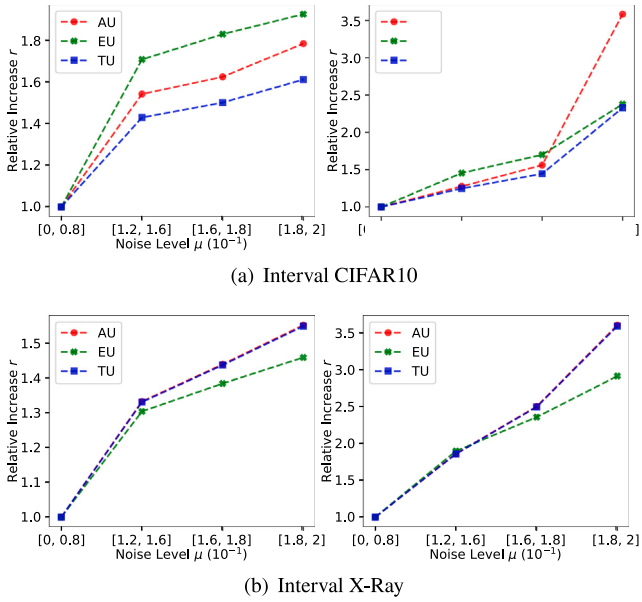


Fig. 10. Relative increase of AU, EU, and TU estimates on interval test instances using different noise levels μ and brightness levels β.

(ii) In instances of interval input data, the proposed models have exhibited the capacity for effective uncertainty quantification.

(iii) The successful integration of ResNet-based CreINNs has shown the efficacy of the proposed Interval Batch Normalization (IBN). The IBN could potentially contribute to realizing conventional INNs in complex neural network architectures.

One of our future research endeavors is to enhance the computational efficiency of CreINNs, aiming to improve inference speed and reduce memory consumption for future practical applications. As CreINNs have indicated a promising capacity for uncertainty quantification, we are also engaged in ongoing efforts to investigate the potential applications of CreINNs in industrial or medical image analysis contexts involving standard and interval data.

CRediT authorship contribution statement

Kaizheng Wang: Writing – original draft, Visualization, Validation, Methodology, Investigation, Conceptualization. **Keivan Shariatmadar:** Writing – review & editing, Supervision. **Shireen Kudukkil Manchingal:** Writing – review & editing, Investigation. **Fabio Cuzzolin:** Writing – review & editing, Supervision. **David Moens:** Writing – review & editing, Supervision. **Hans Hallez:** Writing – review & editing, Supervision.

Declaration of competing interest

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

Acknowledgments

This project has received funding from the European Horizon 2020 research and innovation program under the FET Open grant agreement No. 964505 (E-pi).

Appendix A. Mathematical discussions

A.1. Smoothness of forward propagation

By analyzing all potential results of $[\underline{\boldsymbol{W}}, \overline{\boldsymbol{W}}] \odot [\underline{\boldsymbol{a}}, \overline{\boldsymbol{a}}]$ under various conditions of $\underline{\boldsymbol{a}}$, $\overline{\boldsymbol{a}}$, $\underline{\boldsymbol{W}}$, and $\overline{\boldsymbol{W}}$ (negativity/positivity), the interval multiplication in Eq. (2), denoted as $[\underline{o}, \overline{o}] := [\underline{\boldsymbol{W}}, \overline{\boldsymbol{W}}] \odot [\underline{\boldsymbol{a}}, \overline{\boldsymbol{a}}]$, can be reformulated as follows:

$$\begin{aligned}
\underline{o} &= \min\{\overline{\boldsymbol{W}}, \boldsymbol{0}\} \min\{\overline{\boldsymbol{a}}, \boldsymbol{0}\} + \max\{\underline{\boldsymbol{W}}, \boldsymbol{0}\} \max\{\underline{\boldsymbol{a}}, \boldsymbol{0}\} \\
&\quad + \min\{\max\{\overline{\boldsymbol{W}}, \boldsymbol{0}\} \min\{\underline{\boldsymbol{a}}, \boldsymbol{0}\} - \min\{\underline{\boldsymbol{W}}, \boldsymbol{0}\} \\
&\quad \max\{\overline{\boldsymbol{a}}, \boldsymbol{0}\}, \boldsymbol{0}\} + \min\{\underline{\boldsymbol{W}}, \boldsymbol{0}\} \max\{\overline{\boldsymbol{a}}, \boldsymbol{0}\} \\
\overline{o} &= \min\{\overline{\boldsymbol{W}}, \boldsymbol{0}\} \max\{\underline{\boldsymbol{a}}, \boldsymbol{0}\} + \max\{\underline{\boldsymbol{W}}, \boldsymbol{0}\} \min\{\overline{\boldsymbol{a}}, \boldsymbol{0}\} \\
&\quad + \max\{\min\{\underline{\boldsymbol{W}}, \boldsymbol{0}\} \min\{\underline{\boldsymbol{a}}, \boldsymbol{0}\} - \max\{\overline{\boldsymbol{W}}, \boldsymbol{0}\} \\
&\quad \max\{\overline{\boldsymbol{a}}, \boldsymbol{0}\}, \boldsymbol{0}\} + \max\{\overline{\boldsymbol{W}}, \boldsymbol{0}\} \max\{\overline{\boldsymbol{a}}, \boldsymbol{0}\}
\end{aligned} \tag{A.1}$$

It can be observed that the min or max operation in Eq. (A.1) are continuous, although they are not strictly differentiable at zeros. As a result, the smoothness of the forward propagation of CreINNs ensures that parameter updates are attainable in the same way of automatic differentiation as standard neural networks (Oala et al., 2021).

A.2. Numerical example showing the infeasibility of classical SoftMax

The traditional SoftMax activation function cannot be used to generate valid probability intervals in CreINNs when computing $[\underline{q}, \overline{q}]$ as $\underline{q} = \mathrm{SoftMax}(\underline{a})$ and $\overline{q} = \mathrm{SoftMax}(\overline{a})$, respectively. For instance, assuming that intervals scores are $\underline{a} := (0, -1, 1)^T$ and $\overline{a} := (1, 0, 3)^T$, the $\underline{q}$ and $\overline{q}$ can be computed from SoftMax as

$$\begin{aligned}
\underline{q} &= \mathrm{SoftMax}(\underline{a}) = (0.2447, 0.0900, 0.6653)^T \\
\overline{q} &= \mathrm{SoftMax}(\overline{a}) = (0.1142, 0.0420, 0.8438)^T
\end{aligned}. \tag{A.2}$$

The numerical example shows that the 'probability intervals' are not properly defined as some upper bounds are considerably smaller than the lower bounds.

A.3. Mathematical proofs for interval SoftMax

In the case of binary classification, the Interval Softmax in Eq. (7) reduces to the Sigmoid activation, as follows:

$$\underline{q} = 1 / \left(1 + \exp(\underline{a}^L)\right), \quad \overline{q} = 1 / \left(1 + \exp(\overline{a}^L)\right). \tag{A.3}$$

We can prove that Interval SoftMax can result in valid probability intervals and satisfy the constraint in Eq. (6) for defining a nonempty credal, as follows:

$$\begin{aligned}
\sum_k^C \underline{q}_k &= \sum_k^C \frac{\exp(\underline{a}_k^L)}{\exp(\underline{a}_k^L) + \sum_{j \neq k}^C \exp(\frac{\underline{a}_j^L + \overline{a}_j^L}{2})} \\
&\leq \sum_k^C \frac{\exp(\frac{\underline{a}_k^L + \overline{a}_k^L}{2})}{\exp(\frac{\underline{a}_k^L + \overline{a}_k^L}{2}) + \sum_{j \neq k}^C \exp(\frac{\underline{a}_j^L + \overline{a}_j^L}{2})} = 1 \\
&\leq \sum_k^C \frac{\exp(\overline{a}_k^L)}{\exp(\overline{a}_k^L) + \sum_{j \neq k}^C \exp(\frac{\underline{a}_j^L + \overline{a}_j^L}{2})} = \sum_k^C \overline{q}_k.
\end{aligned} \tag{A.4}$$

Interval SoftMax demonstrates smoothness for backward propagation. The relative partial derivatives can be derived as follows:

$$\frac{\partial \underline{q}_k}{\partial \underline{a}_j^L} = \begin{cases} \underline{q}_k(1 - \underline{q}_k), & k = j \\ -\frac{1}{2} \underline{q}_k \frac{\exp(\frac{\underline{a}_j^L + \overline{a}_j^L}{2})}{\exp(\underline{a}_k^L) + \sum_{j \neq k}^C \exp(\frac{\underline{a}_j^L + \overline{a}_j^L}{2})}, & k \neq j \end{cases} \tag{A.5}$$

$$\frac{\partial \underline{q}_k}{\partial \overline{a}_j^L} = \begin{cases} 0, & k = j \\ -\frac{1}{2} \underline{q}_k \frac{\exp(\frac{\underline{a}_j^L + \overline{a}_j^L}{2})}{\exp(\underline{a}_k^L) + \sum_{j \neq k}^C \exp(\frac{\underline{a}_j^L + \overline{a}_j^L}{2})}, & k \neq j \end{cases} \tag{A.6}$$

$$\frac{\partial \overline{q}_k}{\partial \overline{a}_j^L} = \begin{cases} \overline{q}_k(1 - \overline{q}_k), & k = j \\ -\frac{1}{2} \overline{q}_k \frac{\exp(\frac{\underline{a}_j^L + \overline{a}_j^L}{2})}{\exp(\underline{a}_k^L) + \sum_{j \neq k}^C \exp(\frac{\underline{a}_j^L + \overline{a}_j^L}{2})}, & k \neq j \end{cases} \tag{A.7}$$

$$\frac{\partial \overline{q}_k}{\partial \underline{a}_j^L} = \begin{cases} 0, & k = j \\ -\frac{1}{2} \overline{q}_k \frac{\exp(\frac{\underline{a}_j^L + \overline{a}_j^L}{2})}{\exp(\underline{a}_k^L) + \sum_{j \neq k}^C \exp(\frac{\underline{a}_j^L + \overline{a}_j^L}{2})}, & k \neq j \end{cases} \tag{A.8}$$

The property of "set constraint" remains satisfied in Interval SoftMax. Namely, for any $a_k^L \in [\underline{a}_k, \overline{a}_k]^L$, the condition consistently holds as follows:

$$q_k = \frac{\exp(a_k^L)}{\exp(a_k^L) + \sum_{j \neq k}^C \exp(\frac{\underline{a}_j^L + \overline{a}_j^L}{2})} \in [\underline{q}_k, \overline{q}_k]. \tag{A.9}$$

A.4. Ensemble of probability intervals

It can be proved that the averaged probability intervals $[\underline{\boldsymbol{q}}_{\mathrm{avg}}, \overline{\boldsymbol{q}}_{\mathrm{avg}}]$ for ensemble of CreINNs in Eq. (15) is guaranteed to generate a non-empty credal set, as follows:

$$\sum_k^C \underline{q}_{\mathrm{avg}_k} = \frac{1}{M} \sum_m^M \sum_k^C \underline{q}_{m_k}^* \leq 1 \leq \frac{1}{M} \sum_m^M \sum_k^C \overline{q}_{m_k}^* = \sum_k^C \overline{q}_{\mathrm{avg}_k}. \tag{A.10}$$

A.5. Uncertainty estimation in BNNs and DEs

Given an instance $\boldsymbol{x}$, the prediction of BNNs applying BMA and DEs can be obtained as follows:

$$\tilde{\boldsymbol{q}} := \frac{1}{M} \sum_m^M h_m(\boldsymbol{x}) = \frac{1}{M} \sum_m^M \boldsymbol{q}_m, \tag{A.11}$$

where M is the number of samples used to approximate the posterior distribution of the parameters in BNNs during inference or the number of ensemble members in DEs. $p(\boldsymbol{\omega}|\mathbb{D})$, during inference. h_m denotes the deterministic model sampled from the posterior distribution of BNNs or the mth SNN in DEs. $\boldsymbol{q}_m$ represents the mth single probability prediction.

Employing Shannon entropy as the uncertainty measure, one can approximate the TU of BNNs and DEs as TU $:= H(\tilde{\boldsymbol{q}})$. The AU can be estimated by averaging the Shannon entropy of each single model prediction (Hüllermeier & Waegeman, 2021):

$$\mathrm{AU} := \frac{1}{M} \sum_{m=1}^M H(\boldsymbol{q}_m). \tag{A.12}$$

Consequently, the EU can be disaggregated from TU by EU = TU − AU (Depeweg, Hernandez-Lobato, Doshi-Velez, & Udluft, 2018). In some literature, the EU is interpreted as an approximation of the "mutual information" (Hüllermeier et al., 2022; Hüllermeier & Waegeman, 2021).

Appendix B. Additional experiments

In this Appendix, we evaluate the uncertainty estimation of CreINNs when the model is trained on standard input data but evaluated using internal data. Multiclass and binary cases are considered. Since the baseline models presented in Section 4.1 do not support interval input data, only CreINNs and the CreINN ensemble are evaluated for uncertainty estimation. The models used in this evaluation were trained on the standard CIFAR-10 and X-ray datasets, following the training procedures outlined in Section 4.1.

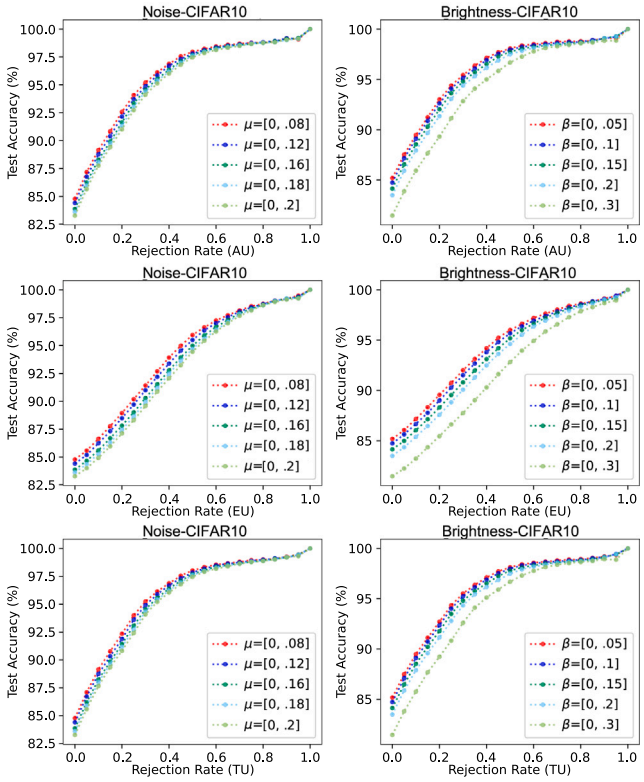


Fig. B.1. AR curves using AU, EU, and TU estimates in multiple cases where the input intervals are constructed from CIFAR10 using different noise levels μ and brightness levels β. The results are averaged over 15 experimental runs.

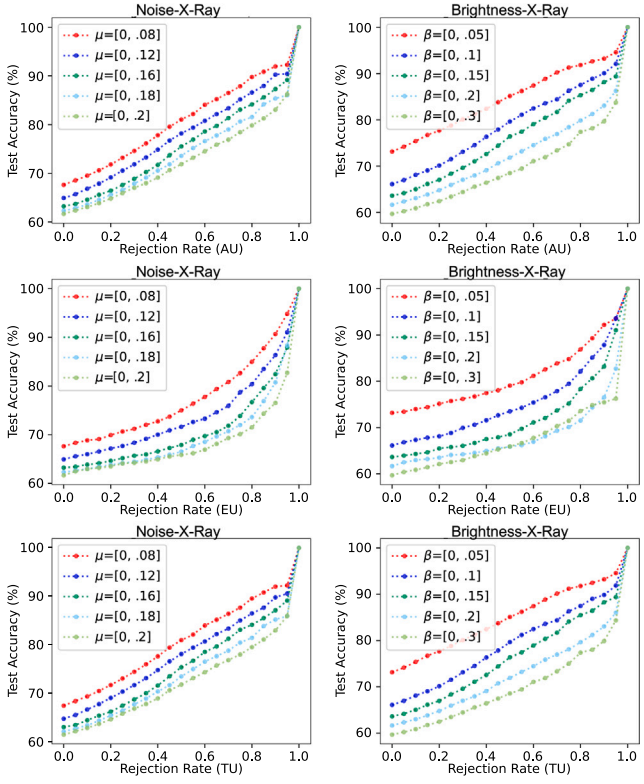


Fig. B.2. AR curves using AU, EU, and TU estimates in multiple cases where the input intervals are constructed from Xray using different noise levels μ and brightness levels β. The results are averaged over 15 experimental runs.

For the interval test data, we apply the data preparation methodology outlined in Section 4.2.1 to generate the interval-based CIFAR10 and X-ray datasets. This process incorporates considerations of noise level (μ) and brightness condition (β). More specifically, the design intervals of μ and β, denoted as $[\mu_1, \mu_2]$ and $[\beta_1, \beta_2]$, were selected as follows:

$$\begin{matrix}[\mu_1, \mu_2] = [0, .08], [0, .12], [0, .16], [0, .18], [0, .2] \\ [\beta_1, \beta_2] = [0, .05], [0, .1], [0, .15], [0, .2], [0, .3]\end{matrix}. \tag{B.1}$$

Figs. B.1 and B.2 demonstrate the AR curves of the CreINN Ensemble in multiple cases where the interval instances are constructed from the CIFAR10 and X-ray datasets using different noise levels μ and brightness levels β. The positive correlation between accuracy and rejection rate in each case verifies the effectiveness of uncertainty quantification. The evidence demonstrates the capability of CreINNs to effectively estimate uncertainty when the model is trained with standard input data and employed for interval input data.

Data availability

The datasets used in this study are publicly available online. The main experimental implementation code is provided at https://github.com/WangKaizheng/CreINNs.

References

Abdar, M., Pourpanah, F., Hussain, S., Rezazadegan, D., Liu, L., Ghavamzadeh, M., et al. (2021). A review of uncertainty quantification in deep learning: Techniques, applications and challenges. *Information Fusion*, *76*, 243–297.

Abe, T., Buchanan, E. K., Pleiss, G., Zemel, R., & Cunningham, J. P. (2022). Deep ensembles work, but are they necessary? *vol. 35*, In *Advances in neural information processing systems* (pp. 33646–33660).

Abellán, J., Klir, G. J., & Moral, S. (2006). Disaggregated total uncertainty measure for credal sets. *International Journal of General Systems*, *35*(1), 29–44.

Betancourt, D., & Muhanna, R. L. (2022). Interval deep learning for computational mechanics problems under input uncertainty. *Probabilistic Engineering Mechanics*, *70*, Article 103370.

Blundell, C., Cornebise, J., Kavukcuoglu, K., & Wierstra, D. (2015). Weight uncertainty in neural network. In *Proceedings of the international conference on machine learning* (pp. 1613–1622). PMLR.

Cao, Y., Wang, X., Wang, Y., Xu, L., & Wang, Y. (2024). An interval neural network method for identifying static concentrated loads in a population of structures. *Aerospace*, *11*(9), 770.

Caprio, M., Dutta, S., Jang, K. J., Lin, V., Ivanov, R., Sokolsky, O., et al. (2024). Credal Bayesian deep learning. *Transactions on Machine Learning Research*.

Cattaneo, M. E., & Wiencierz, A. (2012). Likelihood-based imprecise regression. *International Journal of Approximate Reasoning*, *53*(8), 1137–1154.

Corani, G., Antonucci, A., & Zaffalon, M. (2012). Bayesian networks with imprecise probabilities: Theory and application to classification. In *Data mining: Foundations and intelligent paradigms: Volume 1: Clustering, association and classification* (pp. 49–93).

Corani, G., & Zaffalon, M. (2008). Learning reliable classifiers from small or incomplete data sets: The Naive credal classifier 2.. *Journal of Machine Learning Research*, *9*(4).

Cuzzolin, F. (2009). Credal semantics of Bayesian transformations in terms of probability intervals. *IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics)*, *40*(2), 421–432.

Cuzzolin, F. (2022). The intersection probability: betting with probability intervals. arXiv preprint arXiv:2201.01729.

De Campos, L. M., Huete, J. F., & Moral, S. (1994). Probability intervals: A tool for uncertain reasoning. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, *02*(02), 167–196.

Depeweg, S., Hernandez-Lobato, J.-M., Doshi-Velez, F., & Udluft, S. (2018). Decomposition of uncertainty in Bayesian deep learning for efficient and risk-sensitive learning. In *International conference on machine learning* (pp. 1184–1193). PMLR.

Faza, G. A., Shariatmadar, K., Hallez, H., & Moens, D. (2024). Interval reduced order surrogate modelling framework for uncertainty quantification. In *AIAA scitech 2024 forum* (p. 0387).

Fort, S., & Jastrzebski, S. (2019). Large scale structure of neural network loss landscapes. *vol. 32*, In *Advances in neural information processing systems*.

Gal, Y., & Ghahramani, Z. (2016). Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. In *Proceedings of the international conference on machine learning* (pp. 1050–1059). PMLR.

Garczarczyk, Z. (2000). Interval neural networks. *vol. 3*, In *Proceedings of the IEEE international symposium on circuits and systems* (pp. 567–570). vol.3.

Gawlikowski, J., Tassi, C. R. N., Ali, M., Lee, J., Humt, M., Feng, J., et al. (2023). A survey of uncertainty in deep neural networks. *Artificial Intelligence Review*, *56*(Suppl 1), 1513–1589.

Glorot, X., & Bengio, Y. (2010). Understanding the difficulty of training deep feedforward neural networks. In Y. W. Teh, & M. Titterington (Eds.), *Proceedings of machine learning research*: *vol. 9*, *Proceedings of the thirteenth international conference on artificial intelligence and statistics* (pp. 249–256). Chia Laguna Resort, Sardinia, Italy: PMLR.

Gustafsson, F. K., Danelljan, M., & Schon, T. B. (2020). Evaluating scalable Bayesian deep learning methods for robust computer vision. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops* (pp. 318–319).

He, B., Lakshminarayanan, B., & Teh, Y. W. (2020). Bayesian deep ensembles via the neural tangent kernel. *vol. 33*, In *Advances in neural information processing systems* (pp. 1010–1022).

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770–778).

Hendrycks, D., & Dietterich, T. (2019). Benchmarking neural network robustness to common corruptions and perturbations. In *International conference on learning representations*.

Hendrycks, D., & Gimpel, K. (2017). A baseline for detecting misclassified and out-of-distribution examples in neural networks. In *International conference on learning representations*.

Hickey, T., Ju, Q., & Van Emden, M. H. (2001). Interval arithmetic: From principles to implementation. *Journal of the ACM*, *48*(5), 1038–1068.

Hobbhahn, M., Kristiadi, A., & Hennig, P. (2022). Fast predictive uncertainty for classification with Bayesian deep networks. In *Uncertainty in artificial intelligence* (pp. 822–832). PMLR.

Hoffman, M. D., Gelman, A., et al. (2014). The No-U-Turn sampler: adaptively setting path lengths in Hamiltonian Monte Carlo. *Journal of Machine Learning Research*, *15*(1), 1593–1623.

Hühn, J. C., & Hüllermeier, E. (2008). FR3: A fuzzy rule learner for inducing reliable classifiers. *IEEE Transactions on Fuzzy Systems*, *17*(1), 138–149.

Hüllermeier, E., Destercke, S., & Shaker, M. H. (2022). Quantification of credal uncertainty in machine learning: A critical analysis and empirical comparison. In *Proceedings of machine learning research*: *vol. 180*, *Proceedings of the thirty-eighth conference on uncertainty in artificial intelligence* (pp. 548–557). PMLR.

Hüllermeier, E., & Waegeman, W. (2021). Aleatoric and epistemic uncertainty in machine learning: An introduction to concepts and methods. *Machine Learning*, *110*(3), 457–506.

Ioffe, S., & Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the international conference on machine learning* (pp. 448–456). PMLR.

Ishibuchi, H., Tanaka, H., & Okada, H. (1993). An architecture of neural networks with interval weights and its application to fuzzy regression analysis. *Fuzzy Sets and Systems*, *57*(1), 27–39.

Jaeger, P. F., Lüth, C. T., Klein, L., & Bungert, T. J. (2023). A call to reflect on evaluation practices for failure detection in image classification. In *The eleventh international conference on learning representations*.

Jospin, L. V., Laga, H., Boussaid, F., Buntine, W., & Bennamoun, M. (2022). Hands-on Bayesian neural networks—A tutorial for deep learning users. *IEEE Computational Intelligence Magazine*, *17*(2), 29–48.

Jurio, A., Pagola, M., Mesiar, R., Beliakov, G., & Bustince, H. (2011). Image magnification using interval information. *IEEE Transactions on Image Processing*, *20*(11), 3112–3123.

Kermany, D. S., Goldbaum, M., Cai, W., Valentim, C. C., Liang, H., Baxter, S. L., et al. (2018). Identifying medical diagnoses and treatable diseases by image-based deep learning. *Cell*, *172*(5), 1122–1131.

Khosravi, A., Nahavandi, S., Creighton, D., & Atiya, A. F. (2011). Lower upper bound estimation method for construction of neural network-based prediction intervals. *IEEE Transactions on Neural Networks*, *22*(3), 337–346.

Kingma, D. P., & Welling, M. (2013). Auto-encoding variational Bayes. arXiv preprint arXiv:1312.6114.

Kowalski, P. A., & Kulczycki, P. (2017). Interval probabilistic neural network. *Neural Computing and Applications*, *28*(4), 817–834.

Krizhevsky, A., Nair, V., & Hinton, G. (2009). CIFAR-10 (Canadian Institute For Advanced Research). URL https://www.cs.toronto.edu/~kriz/cifar.html.

Lai, Y., Shi, Y., Han, Y., Shao, Y., Qi, M., & Li, B. (2022). Exploring uncertainty in regression neural networks for construction of prediction intervals. *Neurocomputing*, *481*, 249–257.

Lakshminarayanan, B., Pritzel, A., & Blundell, C. (2017). Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in Neural Information Processing Systems*, *30*.

Lambrou, A., Papadopoulos, H., & Gammerman, A. (2010). Reliable confidence measures for medical diagnosis with evolutionary algorithms. *IEEE Transactions on Information Technology in Biomedicine*, *15*(1), 93–99.

Levi, I. (1980). *The enterprise of knowledge: An essay on knowledge, credal probability, and chance*. MIT Press.

Lienen, J., Demir, C., & Hullermeier, E. (2023). Conformal credal self-supervised learning. In *Conformal and probabilistic prediction with applications* (pp. 214–233). PMLR.

Liu, J., Lin, Z., Padhy, S., Tran, D., Bedrax Weiss, T., & Lakshminarayanan, B. (2020). Simple and principled uncertainty estimation with deterministic deep learning via distance awareness. *Advances in Neural Information Processing Systems*, *33*, 7498–7512.

Molchanov, D., Ashukha, A., & Vetrov, D. (2017). Variational dropout sparsifies deep neural networks. In *Proceedings of the international conference on machine learning* (pp. 2498–2507). PMLR.

Mucsányi, B., Kirchhof, M., & Oh, S. J. (2024). Benchmarking uncertainty disentanglement: Specialized uncertainties for specialized tasks. In *The thirty-eight conference on neural information processing systems datasets and benchmarks track*.

Mukhoti, J., Kirsch, A., van Amersfoort, J., Torr, P. H., & Gal, Y. (2023). Deep deterministic uncertainty: A new simple baseline. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* CVPR, (pp. 24384–24394).

Neal, R. M., et al. (2011). MCMC using Hamiltonian dynamics. *vol. 2*, In *Handbook of Markov Chain Monte Carlo* (p. 2).

Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., & Ng, A. Y. (2011). Reading digits in natural images with unsupervised feature learning.

Oala, L., Heiß, C., Macdonald, J., März, M., Kutyniok, G., & Samek, W. (2021). Detecting failure modes in image reconstructions with interval neural network uncertainty. *International Journal of Computer Assisted Radiology and Surgery*, *16*(12), 2089–2097.

Ovadia, Y., Fertig, E., Ren, J., Nado, Z., Sculley, D., Nowozin, S., et al. (2019). Can you trust your model's uncertainty? Evaluating predictive uncertainty under dataset shift. *vol. 32*, In *Advances in neural information processing systems*.

Pearce, T., Brintrup, A., Zaki, M., & Neely, A. (2018). High-quality prediction intervals for deep learning: A distribution-free, ensembled approach. In *International conference on machine learning* (pp. 4075–4084). PMLR.

Rudner, T. G., Chen, Z., Teh, Y. W., & Gal, Y. (2022). Tractable function-space variational inference in Bayesian neural networks. *Advances in Neural Information Processing Systems*, *35*, 22686–22698.

S. Salem, T., Langseth, H., & Ramampiaro, H. (2020). Prediction intervals: Split normal mixture from quality-driven deep ensembles. In J. Peters, & D. Sontag (Eds.), *Proceedings of machine learning research*: *vol. 124*, *Proceedings of the 36th conference on uncertainty in artificial intelligence* (pp. 1179–1187). PMLR.

Sadeghi, J., De Angelis, M., & Patelli, E. (2019). Efficient training of interval neural networks for imprecise training data. *Neural Networks*, *118*, 338–351.

Sale, Y., Caprio, M., & Hüllermeier, E. (2023). Is the volume of a credal set a good measure for epistemic uncertainty? In *Proceedings of machine learning research*, *Proceedings of the thirty-ninth conference on uncertainty in artificial intelligence* (pp. 1795–1804). PMLR.

Song, Y., & Deng, Y. (2019). Divergence measure of belief function and its application in data fusion. *IEEE Access*, *7*, 107465–107472.

Soubaras, H. (2011). Towards an axiomatization for the generalization of the Kullback-Leibler divergence to belief functions. In *Proceedings of the 7th conference of the European society for fuzzy logic and technology* (pp. 1090–1097). Atlantis Press.

Tretiak, K., Schollmeyer, G., & Ferson, S. (2023). Neural network model for imprecise regression with interval dependent variables. *Neural Networks*, *161*, 550–564.

Vovan, T., Phamtoan, D., Tuan, L. H., & Nguyentrang, T. (2021). An automatic clustering for interval data using the genetic algorithm. *Annals of Operations Research*, *303*, 359–380.

Wen, Y., Vicol, P., Ba, J., Tran, D., & Grosse, R. (2018). Flipout: Efficient pseudo-independent weight perturbations on mini-batches. In *Proceedings of the international conference on learning representations*.

Zaffalon, M. (2002). The naive credal classifier. *Journal of Statistical Planning and Inference*, *105*(1), 5–21.