EPISTEMIC WRAPPING FOR UNCERTAINTY QUANTIFICATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Uncertainty estimation is pivotal in machine learning, especially for classification tasks, as it improves the robustness and reliability of models. We introduce a novel 'Epistemic Wrapping' methodology aimed at improving uncertainty estimation in classification. Our approach uses Bayesian Neural Networks (BNNs) as a baseline and transforms their outputs into belief function posteriors, effectively capturing epistemic uncertainty and offering an efficient and general methodology for uncertainty quantification. Comprehensive experiments employing various BNN baselines and an Interval Neural Network for inference on the MNIST, Fashion-MNIST, CIFAR-10 and CIFAR-100 datasets demonstrate that our Epistemic Wrapper significantly enhances generalisation and uncertainty quantification.

1 INTRODUCTION

In the realm of machine learning, particularly in classification tasks, uncertainty estimation plays a crucial role in enhancing the robustness and reliability of models (Sale et al., 2023). Accurately quantifying uncertainty is vital for applications where decisions must be made with confidence, such as in medical diagnosis (Lambrou et al., 2010), autonomous driving (Fort & Jastrzebski, 2019) and financial forecasting. Traditional deterministic neural networks, while powerful, cannot often effectively capture and express uncertainty (Liu et al., 2020). This shortfall has spurred interest in probabilistic approaches, with Bayesian neural networks (BNNs) emerging as a promising solution in this context. BNNs offer a principled approach to uncertainty estimation by incorporating prior distributions over the model parameters, leading to posterior distributions that reflect model uncertainty (Jospin et al., 2022). Despite their theoretical appeal, BNNs face practical challenges, including high computational costs and complexity in training.

The literature majors on two sources of uncertainty: *Epistemic* Uncertainty (EU) and *Aleatoric* Uncertainty (AU) (Hüllermeier & Waegeman, 2021; Abdar et al., 2021). Epistemic uncertainty is due to a lack of knowledge about the true model parameters and can be reduced with more data or better models. In contrast, aleatoric uncertainty stems from the inherent randomness in the data generation process and cannot be reduced. Over the years, various studies (Hüllermeier & Waegeman, 2021; Abdar et al., 2021) have recognised that accurately modelling parameter uncertainty can produce a variety of credible network models, which are likely to include the true underlying network model, leading to both better EU estimation and more reliable inference. In particular, *second-order uncertainty* frameworks (including belief functions Cuzzolin (2020)) can be employed to model both EU and AU, effectively expressing 'uncertainty about a prediction's uncertainty' (Hüllermeier & Waegeman, 2021; Sale et al., 2023).

BNNs, as one of the prevalent method for uncertainty estimation, treat all the weights and biases of the network as probability distributions. The prediction of the Neural Network is represented as a second-order distribution, thus representing the probability distribution of distributions (Hüllermeier & Waegeman, 2021). Although effective approximation techniques have been developed, such as variational inference (VI) approaches (Blundell et al., 2015; Gal & Ghahramani, 2016) and sampling methods (Neal et al., 2011; Hoffman et al., 2014), the high computational cost of BNNs during training as well as inference time limit their practical adoption, especially in real-time applications (Abdar et al., 2021).

Recent work shows that epistemic uncertainty (EU) can be better captured by frameworks more general than probability distributions Cuzzolin (2024), including credal sets Levi (1980) and belief

functions Shafer (1976), leading to improved robustness and uncertainty estimation Manchingal & Cuzzolin (2022); Manchingal et al. (2025b; 2023); Chan et al. (2024); Wang et al. (2024b;a); Caprio et al. (2024); Manchingal et al. (2025a).

Still, current efforts model (epistemic) uncertainty in the model's *target* space, rather than its *parameter* space.

This paper proposes *Epistemic Wrapper*, a novel method which, for the first time, models EU in the parameter space via a random set representation by 'wrapping' a learnt Bayesian posterior in the form of a *belief function* (Fig. 1). The methodology unfolds organically, beginning with a Bayesian Neural Network (BNN) and culminating in an Interval Neural Network (INN). From a pre-trained BNN, we obtain posterior distributions with parameters (μ, σ), where Gaussian priors are assumed. These posterior distributions are truncated and we compute continuous belief functions over closed intervals. Through Möbius inversion, these belief functions translate into corresponding mass values. These discrete, normalized mass values are then embedded into a continuous representation by fitting a Dirichlet distribution with parameters estimated via the method of L-moments. The interval-valued nature of Dirichlet sampling motivates the use of an Interval Neural Network (INN), which performs stable and calibrated inference. To our knowledge, no previous work has modelled EU in the parameter space via higher-order uncertainty measures. The motivation behind modelling epistemic uncertainty in the parameter space stems from the idea that parameter uncertainty is a primary source of EU. By representing uncertainty directly in the parameter space before it propagates to predictions, we capture model-level uncertainty in a more principled way. Modelling in the parameter space offers several advantages: (a) It provides a prior-agnostic mechanism to represent epistemic uncertainty, without relying solely on the model's output distribution. (b) It enables structured and interpretable sampling through belief functions and Dirichlet distributions, supporting more stable and calibrated uncertainty estimates via interval-based inference. (c) It can be seamlessly integrated with existing BNNs offering flexibility and broader applicability.

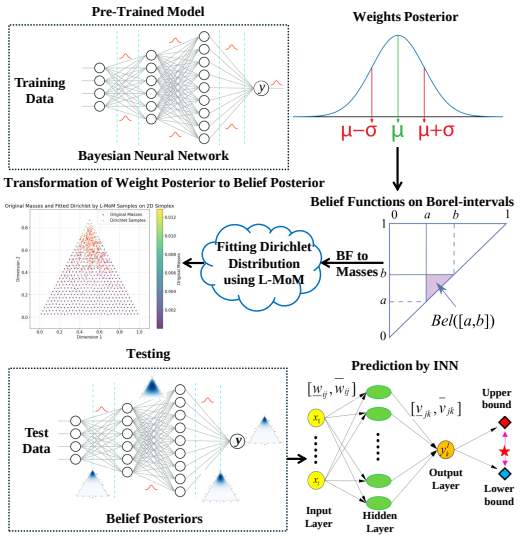


Figure 1: **Epistemic Wrapper** transforms weights posteriors from a Bayesian Neural Network into belief posteriors. It involves extracting probability posteriors, calculating belief values over Borel intervals, computing mass values using Moebius inversion, fitting a Dirichlet distribution to these masses via method of L-moments, and using the resulting belief posteriors as weights to Interval Neural Networks for final predictions.

The strength of our approach lies in the novel, careful and coherent integration of theoretically-grounded techniques. While individual components, such as belief function construction and Dirichlet fitting, are based on existing concepts, their combination into a framework that learns Random Set (RS) representations directly over BNN parameters is new and forms the core contribution of this work. In particular, the likelihood transformation at the core of our approach is not heuristic but is grounded in a set of rigorous rationality axioms. It is in fact the only belief function satisfying these properties, which provides a formal and well-founded basis for this transformation. The subsequent use of the Dirichlet mass function to encode the continuous belief function is equally principled. Our choice is guided by the need for an efficient RS representation on the collection of intervals, where any probability distribution defined over the simplex could theoretically be used. However, Dirichlet distributions have recently demonstrated strong practical effectiveness in modelling epistemic uncertainty in neural networks, as highlighted in evidential learning literature Sensoy et al. (2018). Finally, inference with INNs is also statistically motivated, as it provides an efficient way to propagate belief-based uncertainty into interval predictions. This ensures that every element of our wrapping framework is anchored in rigorous theoretical principles, rather than assembled heuristics.

Our approach leverages the strengths of BNNs while injecting the ability of higher-order measure to improve robustness and uncertainty estimation. The contributions to the literature are therefore: (1) **The first modelling of EU in the parameter space using higher-order uncertainty measures**. (2) A novel and versatile **Epistemic Wrapper** concept, that can be applied to any BNN baseline to convert it automatically into a belief-function posterior. (3) Based on the above, a **novel approach to uncertainty estimation** in classification which efficiently leverages BNNs as a foundation. Our experiments further demonstrate the versatility of the proposed Epistemic Wrapper across multiple datasets. For example, on MNIST, the baseline BNN achieved an accuracy of $72.44\% \pm 0.24$, whereas Epi-Wrapper substantially improved performance to $91.02\% \pm 0.05$. On Fashion-MNIST, the BNN reached $58.91\% \pm 0.24$, while Epi-Wrapper achieved $82.45\% \pm 0.10$. Similar improvements are consistently observed on large-scale benchmarks such as CIFAR-10 and CIFAR-100 with ResNet-18 and VGG-16 backbones, highlighting the effectiveness and scalability of our approach in enhancing predictive performance and reliability across diverse settings.

2 Relevant Work

Epistemic approaches. While various types of uncertainty measures Cuzzolin (2021) have been employed in machine learning in the past Cuzzolin & Gong (2013); Cuzzolin (2018a); Liu et al. (2019); Gong & Cuzzolin (2017), recent advancements in epistemic uncertainty modelling have introduced a range of methods to improve predictive reliability across various neural architectures. Evidential deep learning predicts second-order probability distributions to estimate uncertainty, but faces challenges in optimisation and interpretation (Juergens et al., 2024). Methods like G-ΔUQ refine uncertainty calibration in Graph Neural Networks (GNNs) through stochastic data centring (Trivedi et al., 2024), while SPDE-based GNNs employ Q-Wiener processes for uncertainty propagation in complex graphs (Lin et al., 2024). The Graph Energy-Based Model (GEBM) leverages graph diffusion to quantify uncertainty at different structural levels (Fuchsgruber et al., 2024), and credal set-based ensemble learning constructs plausible probability distributions to measure aleatoric and epistemic uncertainty (Hofman et al., 2024).

Crucially, (Manchingal et al., 2025a) introduces a unified evaluation framework for uncertainty-aware classifiers, mapping all uncertainty-aware predictions into credal sets Cuzzolin (2008a), thus enabling a standardised assessment of epistemic uncertainty across BNNs, Deep Ensembles, Evidential Deep Learning (EDL), and Credal Set-based approaches. (Manchingal et al., 2025b) extends uncertainty modelling through Random-Set Neural Networks (RS-NNs), which employ random set theory to construct belief-based uncertainty representations, providing a more flexible alternative to conventional probabilistic models. Credal Interval Neural Networks Wang et al. (2025), instead, represent predictions as credal sets, which encapsulate a range of probable outcomes, thereby explicitly modelling epistemic uncertainty. Building on the latter, Credal Deep Ensembles (Wang et al., 2024b) predict and aggregate ensembles of convex sets of probability distributions, resulting in a more conservative and informative epistemic uncertainty quantification. In an alternative approach Charpentier et al. (2020); Malinin et al. (2019); Sensoy et al. (2018) predictions are modelled as Dirichlet distributions. A key challenge with these methods is the lack of ground truth labels for uncertainty, making direct supervision difficult.

While these models can be highly effective, they primarily quantify uncertainty at the target level, leaving the question of modelling epistemic uncertainty at parameter level open. In contrast, our proposed Epistemic Wrapper leverages BNNs to do exactly so, by transforming probability posteriors into belief posteriors, to offer a robust solution for uncertainty quantification in classification tasks.

Interval neural networks. Traditional *Interval Neural Networks* (INNs) employ deterministic interval-based representations for inputs, outputs, weights, and biases, ensuring robust uncertainty modelling in neural computations. The forward propagation in an INN follows interval arithmetic principles, where the interval-formed activations in each layer are computed using element-wise interval addition, subtraction, and multiplication (Hickey et al., 2001). Specifically, the activation output of the l^{th} layer is determined by applying a monotonically increasing activation function to the interval-weighted sum of the previous layer's outputs and the corresponding interval biases. This formulation guarantees the *set constraint* property, ensuring that for any given input and network parameters within their defined intervals, the computed activations remain bounded within a well-defined range. When the activation function is non-negative (e.g., ReLU), further simplifications allow efficient computation of interval bounds using minimum and maximum operators. This structured

interval propagation enables INNs to maintain rigorous mathematical constraints while modelling uncertainties in deep learning architectures (Morales & Sheppard, 2025).

Our approach employs INNs at inference time using our epistemic wrapper weights, sampled from the wrapped belief posterior, thus leveraging their structured interval-based representations to quantify and propagate epistemic uncertainty effectively.

3 METHODOLOGY

The proposed *Epistemic Wrapping* provides a principled framework for transforming learned posterior distributions on model parameters into belief-function posteriors. The details are presented below:

3.1 LEARNING A BAYESIAN POSTERIOR (BASELINE)

Epistemic wrapping begins by exploiting BNNs with Gaussian priors parameterized by (μ, σ); training then yields posterior distributions over the weights, also parameterized by (μ, σ). The posterior distribution $p(\boldsymbol{\omega}|\mathbb{D})$ is defined via Bayes' theorem, but is generally intractable. To address this, we employ Variational Inference, approximating $p(\boldsymbol{\omega}|\mathbb{D})$ with a variational distribution $q(\boldsymbol{\omega})$ that is optimized to match the true posterior. At inference time, Bayesian Model Averaging is performed by sampling weights from the variational posterior.

3.2 DYNAMIC TRUNCATION

A typical Bayesian weight posterior will look like the one in Fig. 2. These posterior distributions are then truncated using a *dynamic distribution truncation* mechanism, an adaptive technique that defines the range of a distribution based on its mean and standard deviation. This dynamically scales the bounds to the parameter values according to the variance of the distribution, ensuring tighter truncation for distributions with smaller variances, where the probability mass is more concentrated, and looser bounds for those with larger variances. The truncation bounds are calculated as: Lower Bound $= \mu -$ dynamic_multiplier $\cdot\, \sigma$, Upper Bound $= \mu +$ dynamic_multiplier $\cdot\, \sigma$, where μ is the mean, σ is the standard deviation, and dynamic_multiplier $= \min(5.0, \frac{1.0}{\sigma})$, ensuring that the multiplier decreases for low-variance distributions while capping its value at 5.0 to prevent excessive truncation in high-variance cases. We selected this approach as it provides a balance between capturing the significant probability mass of the distribution and avoiding overly wide or narrow bounds, which could either dilute meaningful mass representation or exclude critical probabilistic regions.

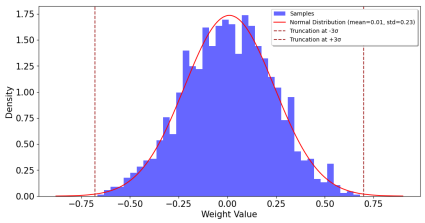


Figure 2: Posterior distribution of weights for the last layer. The histogram displays the sampled weights, overlaid with a fitted normal distribution. Vertical dashed lines indicate truncation at $\pm 3\sigma$.

3.3 CONTINUOUS BELIEF FUNCTIONS ON CLOSED INTERVALS

Belief functions (Cuzzolin, 2014b) can be easily extended to continuous spaces (e.g., a network's parameter space) by defining a continuous mass function over the collection of *closed intervals*, rather than the entire power set. For an introduction to belief functions, see Appendix Section A.1. Given a network parameter ω with values in $\mathbb{R}$, this requires defining a continuous PDF over the collection of intervals $[a, b] \subset \mathbb{R}$ (Cuzzolin, 2020). Here we will assume that parameter values are bounded after truncation (for illustration, in $[0, 1]$); however, the method can be easily extended to unbounded parameter values as well. The space of all closed intervals in $[0, 1]$ is a triangle, as illustrated in Fig. 3. Given a continuous mass function there (non-negative and with integral 1), one can compute

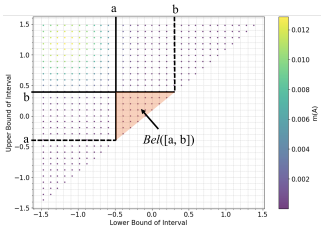


Figure 3: Graphical visualisation of the continuous PDF/mass function over intervals, with the area representing $\mathrm{Bel}([a, b])$.

the belief and plausibility value of a parameter interval $A = [a, b]$ by integrating it over specific regions of the triangle Smets (2005) (Fig. 3). The same applies for parameters bounded by arbitrary values.

Given a truncated posterior distribution over a network's weights, learned by a BNN, our Wrapper transforms it into a continuous belief function using the method proposed in Wasserman (1990). For any closed interval $A = [a, b]$ of the parameter space, one can compute its *plausibility* from the posterior distribution by taking the supremum of the normalised posterior $\hat{p}(\boldsymbol{\omega}|\mathbb{D})$ across all $\boldsymbol{\omega} \in A$, namely:

$$Pl_{\Theta}(A|\mathbb{D}) = \sup_{\omega \in A} \hat{p}(\boldsymbol{\omega}|\mathbb{D}). \tag{1}$$

The corresponding belief value is then calculated as the complement of the plausibility:

$$Bel_{\Theta}(A|\mathbb{D}) = 1 - Pl_{\Theta}(A^c|\mathbb{D}), \tag{2}$$

ultimately providing the sought random-set representation in the parameter space.

The method is grounded into rationality principles, such as (i) the likelihood principle, (ii) compatibility with Bayesian inference (which ensures that combining a Bayesian prior with the belief function yields the Bayesian posterior), and (iii) the principle of Minimum Commitment, which maintains that among the belief functions satisfying the previous two principles, the one chosen should commit to the least amount of information necessary Cuzzolin (2020).

To cap complexity, sample belief values can be computed for a grid of parameter values only. The corresponding mass values can then be easily obtained by Moebius inversion (Shafer, 1976).

3.4 Fitting a Dirichlet Distribution

Epistemic Wrapper employs the method of *L-moments* Hosking (2018) to fit a Dirichlet distribution to the grid of mass values so obtained.

A **Dirichlet distribution** is a family of continuous multivariate probability distributions parameterised by a vector α of positive real numbers; in fact, a multivariate extension of the Beta distribution

$$f(x_1, \ldots, x_K; \alpha_1, \ldots, \alpha_K) = \frac{1}{B(\alpha)} \prod_{i=1}^{K} x_i^{\alpha_i - 1}. \tag{3}$$

As they are defined on the collection of vectors $x \in [0, 1]^K$ of dimension K whose coordinates add to 1, Dirichlet distributions can be interpreted as second-order distributions. For an introduction to Dirichlet distribution, see Appendix Section A.2.

The **method of L-Moments** is a statistical approach employed for parameter estimation in probability distributions. Here we utilise this method to estimate the parameters of a Dirichlet distribution over mass values. L-moments are analogous to conventional moments but are based on linear combinations of order statistics.

To fit a Dirichlet distribution to the grid of mass values, we compute weighted L-moments from the data represented in a 3D simplex space, where each data point has an associated weight derived from its mass value. An example grid in a 2D simplex representation is shown in Fig. 4.

Computation of weighted L-Moments. We first need to compute the first-order and second-order weighted L-moments from the grid of data points. Let $\mathbf{x}_i \in \mathbb{R}^3$ denote the i-th data point in the 3D simplex and w_i its associated weight (derived by normalizing the mass values, so that $\sum_i w_i = 1$). The L-moments are computed as follows. *First-order L-moment (L_1).* The weighted mean of the points in the simplex and is given by:

$$L_1 = \sum_{i=1}^{n} w_i \mathbf{x}_i. \tag{4}$$

Second-order L-moment (L_2). The weighted spread (variance) of the points relative to L_1:

$$L_2 = \frac{\sum_{i=1}^{n} w_i (\mathbf{x}_i - L_1)^2}{\sum_{i=1}^{n} w_i} \tag{5}$$

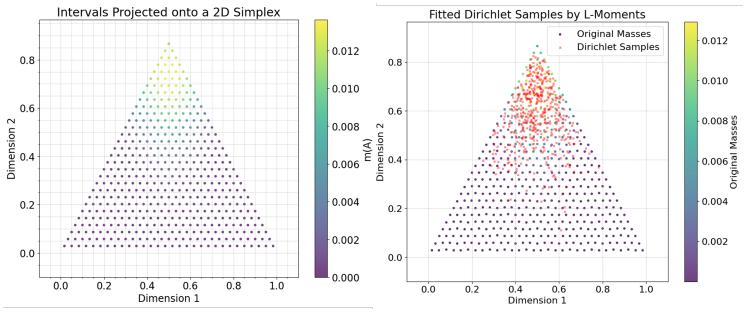


Figure 4: **Left:** Intervals projected onto a 2D simplex. Each point represents an interval $A = [a, b]$ with its location determined by the values a and b, and the colour scale indicates the corresponding mass values $m(A)$, ranging from 0.00 to 0.012. **Right:** Visualization of Dirichlet samples on a 2D simplex. Points sampled from the fitted Dirichlet distribution over mass values.

To ensure numerical stability, a small value ϵ is added to L_2 when necessary, preventing division by zero in subsequent computations.

Using the computed L-moments, we can estimate the parameters $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \alpha_3)$ of the Dirichlet distribution. The relationship between L-moments and the Dirichlet parameters is expressed as:

$$\alpha_k = L_{1,k} \left(\frac{L_{1,k}(1 - L_{1,k})}{L_{2,k}} - 1 \right), \quad k = 1, 2, 3 \tag{6}$$

where $L_{1,k}$ and $L_{2,k}$ are the respective components of the first and second L-moments along each axis of the simplex. Note that $k = 3$ corresponds to the dimensionality of the projected probability simplex. Although equation 6 can, in principle, yield negative α_k values if the second L-moment $L_{2,k}$ exceeds $L_{1,k}(1 - L_{1,k})$ (which may occur in sparse or highly skewed data), our implementation prevents this in practice. We clamp unstable L_2 values, enforce a small positive threshold ϵ on all α_k. These safeguards ensure that the estimated Dirichlet parameters remain valid, positive, and numerically stable across datasets.

Visual representation show that, after fitting a Dirichlet distribution to the grid of mass values, samples of it are also concentrated on the top of the simplex as shown in Fig. 4-Right.

Theoretical Properties of the Epistemic Wrapper. The Epistemic Wrapper preserves an important theoretical property. Specifically, the original Bayesian posterior P lies within the credal set induced by the belief and plausibility functions after wrapping, satisfying

$$Bel(A) \leq P(A) \leq Pl(A) \quad \text{for all measurable sets } A.$$

This relation ensures that our transformation is conservative: it enriches the original posterior with second-order uncertainty without distorting the underlying predictive information. Consequently, the model maintains consistency with the Bayesian posterior while gaining robustness, which helps to explain the observed improvements in generalization and uncertainty estimation. The plausibility $Pl(A)$ captures the maximum value of $\hat{p}(\omega|\mathbb{D})$ over A, while belief $Bel(A)$ captures the minimum guaranteed mass by considering the complement A^c. Since $P(A)$ is the integral of $\hat{p}(\omega|\mathbb{D})$ over A, it must lie between the least conservative estimate (Bel) and the most generous estimate (Pl) over A. This follows from the construction rules of likelihood-based belief functions and random set theory (see (Shafer, 1976; Wasserman, 1990; Cuzzolin, 2020)).

3.5 Inference via Interval Neural Networks

Since Dirichlet sampling yields interval-based representations, the framework integrates naturally with an Interval Neural Network (INN), where weight intervals are derived from a combination of Dirichlet-derived intervals (wrapped parameters) and Gaussian posteriors (unwrapped parameters). While the overall architecture follows a standard INN, our formulation integrates both Dirichlet- and Gaussian-based uncertainty representations within a unified framework. This hybrid design enables stable and well-calibrated uncertainty estimates, particularly for epistemic uncertainty, and ensures smooth interaction between wrapped and unwrapped weights during inference.

Table 1: Classification accuracies on MNIST under different budgeting criteria before and after fine-tuning for posterior weights. Results are from 15 runs. Best scores are presented in bold.

BUDGETING	MLP SIZE	BEFORE FINE-TUNING		AFTER FINE-TUNING		
		INN	EPI-WRAPPER	BNN	INN	EPI-WRAPPER
$\uparrow \sigma$	2	9.11 ± 0.53	$\mathbf{12.93 \pm 0.75}$	33.14 ± 0.22	57.77 ± 0.81	$\mathbf{62.43 \pm 0.46}$
	4	10.44 ± 0.77	$\mathbf{19.94 \pm 0.47}$	40.19 ± 0.55	83.32 ± 0.24	$\mathbf{85.17 \pm 0.10}$
	8	9.33 ± 0.54	$\mathbf{25.46 \pm 1.57}$	72.44 ± 0.24	$\mathbf{91.12 \pm 0.08}$	91.08 ± 0.09
$\uparrow \mu$	2	9.11 ± 0.53	$\mathbf{10.63 \pm 0.34}$	33.14 ± 0.22	57.77 ± 0.81	$\mathbf{63.06 \pm 0.47}$
	4	10.44 ± 0.77	$\mathbf{18.13 \pm 0.71}$	40.19 ± 0.55	83.32 ± 0.24	$\mathbf{85.35 \pm 0.06}$
	8	9.33 ± 0.54	$\mathbf{51.33 \pm 1.21}$	72.44 ± 0.24	$\mathbf{91.12 \pm 0.08}$	91.02 ± 0.05
$\uparrow \mu + \sigma$	2	9.11 ± 0.53	$\mathbf{10.45 \pm 0.16}$	33.14 ± 0.22	57.77 ± 0.81	$\mathbf{63.02 \pm 0.55}$
	4	10.44 ± 0.77	$\mathbf{18.55 \pm 0.68}$	40.19 ± 0.55	83.32 ± 0.24	$\mathbf{85.18 \pm 0.07}$
	8	9.33 ± 0.54	$\mathbf{51.31 \pm 1.29}$	72.44 ± 0.24	91.12 ± 0.08	$\mathbf{91.12 \pm 0.07}$
RANDOM-SELECTION	2	$\mathbf{9.11 \pm 0.53}$	9.80 ± 0.00	33.14 ± 0.22	57.77 ± 0.81	$\mathbf{64.84 \pm 0.16}$
	4	10.44 ± 0.77	$\mathbf{17.35 \pm 0.27}$	40.19 ± 0.55	83.32 ± 0.24	$\mathbf{85.45 \pm 0.06}$
	8	$\mathbf{9.33 \pm 0.54}$	9.23 ± 0.64	72.44 ± 0.24	$\mathbf{91.12 \pm 0.08}$	90.80 ± 0.09

Namely, the unwrapped weights (Gaussian posteriors) generate the following intervals: Lower Bound $= \mu - \sigma$, Upper Bound $= \mu + \sigma$. The process of defining these wrapped and unwrapped weights can be considered the weight initialisation step for the INN model. In contrast, the baseline INN retains the default weight initialisation scheme (Kim, 1993). In other words, at inference time, the model operates with two distinct sets of weights: (i) with random initialisation, (baseline INN) and (ii) with weights transformed by the Epi-Wrapper. Both models are also fine-tuned: the baseline with randomly initialized weights, while the Epi-Wrapper utilizes weights transformed through the wrapping strategy. For a fair comparison, both the baseline and our Epi-Wrapper operate under identical functional settings. For more details on the functionality of INNs, see Appendix Section A.3.

4 EXPERIMENTS

In this section, we evaluate the effectiveness of our Epistemic Wrapper in terms of uncertainty estimation and predictive performance. We summarize the experimental setup, including datasets, model architectures, and ablation studies, and compare our method against relevant baselines. We employed Bayesian baselines including BNNR (Auto-Encoding Variational Bayes (Kingma & Welling, 2013) with the local re-parameterization trick (Molchanov et al., 2017)), and BNNF (Flipout gradient estimator with the negative evidence lower bound loss (Wen et al., 2018)). We use four standard image classification benchmarks: MNIST (LeCun, 1998), Fashion-MNIST (Xiao et al., 2017), CIFAR-10, and CIFAR-100 (Krizhevsky, 2009b). Further implementation details of the Epi-Wrapper algorithm are provided in Appendix B.

4.1 BUDGETING

A budgeting strategy is introduced to selectively transform the posterior distributions of a *subset* of parameters (weights and biases). Posteriors not selected, referred to as *unwrapped* posteriors, retain their original learned parameters. We propose four distinct budgeting strategies: three are parameter-based, prioritizing posteriors with high μ, high σ, or simultaneously high μ and σ, while the fourth employs a random selection strategy that remains unbiased w.r.t. these parameter values.

Ablation on Budgeting. We first conducted an ablation study on the MNIST dataset in which four different Budgeting criterias were tested.

In **Budgeting using High Variance ($\uparrow \sigma$)** we sampled 5% weights with 'High Variance' from the posterior distributions (parameters: μ, σ) of the whole model and transformed them to belief posteriors using Epistemic Wrapper. The results are shown in Table 1, where 'MLP size' is the number of hidden units in the single hidden layer of the model. Since inference in our methodology is done using INNs, we compare our results with those of INN (taken as a baseline). The results shows that using the wrapper improves the quality of the weights initialization with respect to the INN baseline. For instance, an MLP with 32 hidden units and weights randomly initialized achieved an accuracy of 10.37% on the test data, while for our wrapper the test accuracy was 50.20%.

Budgeting using High Mean ($\uparrow \mu$) is another strategy in which we sample and 'wrap' the 5% weights with 'High Mean' from the posterior distributions. From the results shown in Table 1, it can be seen that 'High Mean' performs better for MLP size (no hidden units) $= 8$.

Table 2: Classification accuracies of INN and Epi-Wrapper models before and after fine-tuning across multiple datasets and Bayesian backbones (MLP, LeNet-5, ResNet-18, VGG-16) with BNNF and BNNR baselines. Reported values are mean ± standard deviation over 15 runs.

Dataset	Backbone	# Params	Baseline	BNN	Before Fine-Tuning		After Fine-Tuning	
					INN	Epi-Wrapper	INN	Epi-Wrapper
MNIST	MLP	12.7K	BNNF	72.44 ± 0.24	9.33 ± 0.54	**51.33 ± 1.21**	91.07 ± 0.08	**91.12 ± 0.05**
Fashion-MNIST	MLP	12.7K	BNNF	58.91 ± 0.24	8.57 ± 1.03	**26.93 ± 1.44**	82.41 ± 0.19	**82.45 ± 0.10**
CIFAR-10	LeNet-5	166.3K	BNNF	47.26 ± 0.24	9.80 ± 0.18	**42.34 ± 0.12**	45.92 ± 0.36	**47.89 ± 0.01**
			BNNR	47.09 ± 0.13	9.80 ± 0.18	**42.45 ± 0.20**	45.92 ± 0.36	**47.99 ± 0.09**
	ResNet-18	9.82M	BNNF	86.79 ± 0.21	10.38 ± 0.17	**20.11 ± 0.02**	87.07 ± 0.23	**90.28 ± 0.09**
			BNNR	85.83 ± 0.30	10.38 ± 0.17	**26.71 ± 0.11**	87.07 ± 0.23	**89.96 ± 0.07**
	VGG-16	30.24M	BNNF	87.39 ± 0.67	9.56 ± 0.33	**65.83 ± 0.06**	87.99 ± 0.30	**88.70 ± 0.06**
			BNNR	88.29 ± 0.87	9.56 ± 0.33	**50.31 ± 0.08**	87.99 ± 0.30	**89.87 ± 0.09**
CIFAR-100	ResNet-18	9.8M	BNNF	67.38 ± 0.92	1.23 ± 0.11	**13.29 ± 0.51**	62.70 ± 0.19	**67.41 ± 0.33**
			BNNR	67.47 ± 0.45	1.23 ± 0.11	**19.48 ± 0.05**	62.70 ± 0.19	**67.49 ± 0.32**
	VGG-16	30.24M	BNNF	65.48 ± 0.85	0.96 ± 0.04	**23.32 ± 0.04**	60.10 ± 1.04	**65.55 ± 0.13**
			BNNR	66.59 ± 0.36	0.96 ± 0.04	**16.11 ± 0.05**	60.10 ± 1.04	**66.63 ± 0.19**

Table 3: OoD detection metrics (AUROC, AUPRC, Entropy) for INN (baseline) vs. Epi-Wrapper (ours) across MNIST, Fashion-MNIST, CIFAR-10, and CIFAR-100 datasets using BNNF and BNNR baselines with MLP, LeNet-5, ResNet-18, and VGG-16 backbones.

Dataset	Backbone	# Params	Baseline	AUROC (↑)		AUPRC (↑)		Entropy (↑)	
				INN	Epi-Wrapper	INN	Epi-Wrapper	INN	Epi-Wrapper
MNIST	MLP	12.7K	BNNF	0.532 ± 0.011	**0.667 ± 0.009**	0.895 ± 0.035	**0.912 ± 0.055**	0.201 ± 0.044	**0.287 ± 0.067**
Fashion-MNIST	MLP	12.7K	BNNF	0.605 ± 0.041	**0.690 ± 0.076**	0.855 ± 0.015	**0.922 ± 0.015**	0.247 ± 0.030	**0.299 ± 0.019**
CIFAR-10	LeNet-5	166.3K	BNNF	0.678 ± 0.006	**0.749 ± 0.001**	0.620 ± 0.048	**0.751 ± 0.003**	1.763 ± 0.010	**1.995 ± 0.045**
			BNNR	0.678 ± 0.006	**0.781 ± 0.009**	0.620 ± 0.004	**0.728 ± 0.007**	1.763 ± 0.010	**2.005 ± 0.013**
	ResNet-18	9.82M	BNNF	0.806 ± 0.014	**0.859 ± 0.003**	0.724 ± 0.019	**0.799 ± 0.008**	0.569 ± 0.035	**0.659 ± 0.034**
			BNNR	0.806 ± 0.014	**0.861 ± 0.005**	0.724 ± 0.019	**0.824 ± 0.008**	0.569 ± 0.035	**0.764 ± 0.021**
	VGG-16	30.24M	BNNF	0.849 ± 0.011	**0.850 ± 0.011**	0.796 ± 0.014	**0.801 ± 0.015**	0.606 ± 0.058	**0.611 ± 0.046**
			BNNR	0.849 ± 0.011	**0.856 ± 0.010**	0.796 ± 0.014	**0.799 ± 0.015**	0.606 ± 0.058	**0.601 ± 0.034**
CIFAR-100	ResNet-18	9.8M	BNNF	0.616 ± 0.031	**0.705 ± 0.089**	0.688 ± 0.005	**0.770 ± 0.040**	1.908 ± 0.043	**1.912 ± 0.067**
			BNNR	0.616 ± 0.031	**0.690 ± 0.019**	0.688 ± 0.005	**0.710 ± 0.029**	1.908 ± 0.043	**1.920 ± 0.075**
	VGG-16	30.24M	BNNF	0.576 ± 0.001	**0.775 ± 0.009**	0.548 ± 0.005	**0.789 ± 0.022**	1.540 ± 0.018	**1.549 ± 0.090**
			BNNR	0.576 ± 0.001	**0.755 ± 0.003**	0.548 ± 0.005	**0.718 ± 0.005**	1.540 ± 0.018	**1.543 ± 0.069**

In Budgeting using High Mean and High Variance (↑ (μ, σ)) we rank the parameters by computing a combined score, defined as the sum of the mean and variance of their posterior distributions: combined_score $= \mu + \sigma$. This acts as a proxy for an upper bound of the posterior distribution, allowing us to prioritize parameters that are either highly informative (high mean) or uncertain (high variance). We then wrap these top 5% weights using Epi-wrapper. The results are shown in Table 1. This strategy allows us to selectively wrap the most influential and uncertain parameters, ensuring that the transformation captures meaningful epistemic uncertainty. However, this approach also imposes a strict constraint on the selection process, as only weights satisfying both conditions are chosen, which may limit flexibility in certain scenarios.

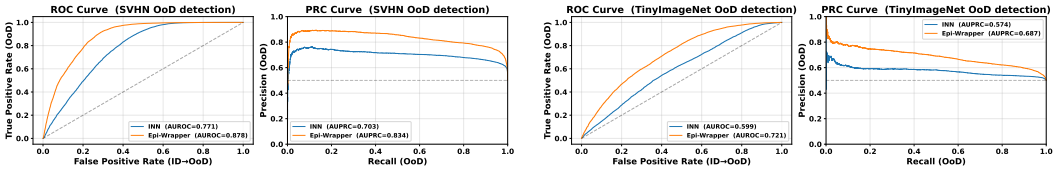
Budgeting using Random Selection (μ, σ) is done by randomly selecting 5% weights from the baseline BNN and extract belief posteriors using the wrapper. Table 1 shows that the results are worse than with other strategies. This is due to the fact that random sampling, while giving us an unbiased selection of posterior weights, may miss those posterior distributions with high uncertainty that can be improved using our wrapping approach.

4.2 Fine-Tuning

We performed the fine-tuning of the models, the INN (baseline) and Epi-Wrapper (ours), on the training data. The results are shown in Table 1 and Table 2. In comparison to the INN and BNN baselines, our model performs well as the wrapping of weights acts as an initialization strategy in fine-tuning. The details of Fine-Tuning are presented in Appendix Section C.2

4.3 iD and OoD Experimental Evaluation

Uncertainty via Predictive Entropy and OoD Detection. Our interval network outputs, for each input x, a pair of class–logit vectors $(\ell^{\mathrm{L}}, \ell^{\mathrm{U}}) \in \mathbb{R}^C$ representing lower and upper logits. We form a sin-



(a) ROC and PRC curves for iD CIFAR-10 vs. SVHN OoD detection using ResNet-18 with BNNR (Molchanov et al., 2017).

(b) ROC and PRC curves for iD CIFAR-100 vs. TinyImageNet OoD detection using VGG-16 with BNNR (Molchanov et al., 2017).

Figure 5: ROC and PRC results for OoD detection on SVHN and TinyImageNet benchmarks using INN and Epi-Wrapper. The curves illustrate performance trade-offs under entropy-based scoring, with AUROC and AUPRC values reported in the legends.

gle logit vector by midpoint aggregation $\tilde{\ell} = \frac{1}{2}\big(\ell^{\mathrm{L}} + \ell^{\mathrm{U}}\big)$, and convert to class probabilities with temperature T (default $T{=}1$) via $p_T(y{=}c \,|\, x) = \mathrm{softmax}\big(\tilde{\ell}/T\big)_c$. We quantify predictive uncertainty using the (Shannon) predictive entropy $H_T(x) = -\sum_{c=1}^{C} p_T(y{=}c \,|\, x) \, \log p_T(y{=}c \,|\, x)$. For the out-of-distribution (OoD) set $\mathcal{D}_{\text{ood}}$ we report the mean OoD entropy $\overline{H}_T^{\text{OoD}} = |\mathcal{D}_{\text{ood}}|^{-1} \sum_{x \in \mathcal{D}_{\text{ood}}} H_T(x)$, and analogously compute the mean entropy on the in-distribution (ID) test set. We further assess OoD separability by using $H_T(x)$ as a scalar score (higher implies more OoD-like). Concatenating the ID and OoD scores with labels $\{0, 1\}$ (ID as 0, OoD as 1), we compute the area under the ROC curve (AUROC) and the area under the precision–recall curve (AUPRC). Thus, in our experiments *entropy is the uncertainty measure*: higher entropy indicates greater predictive uncertainty and is used directly to evaluate both calibration (through mean entropy and NLL/ECE) and OoD detection (via AUROC/AUPRC). Table 3 shows that Epi-Wrapper consistently improves OoD detection performance across all datasets and architectures. In particular, it achieves higher AUROC, AUPRC, and entropy scores compared to the INN baseline, demonstrating stronger separability between iD and OoD samples. This trend is further confirmed by the ROC and PRC curves in Figure 5a and Figure 5b, where Epi-Wrapper outperforms the INN baseline. Additional ROC/PRC results are presented in Section C.4, and calibration results (NLL/ECE) are reported in Appendix C.5.

4.4 Additional Results and Details in Appendix

We summarize the contents of the Appendix in this section. Appendix A provides theoretical background concepts, including Belief functions, Dirichlet distributions, and the functionality of Interval Neural Networks (INNs). Appendix B contains implementation details, dataset descriptions, and Bayesian baseline setups. It also specifies the backbone architectures (MLP, LeNet-5, ResNet-18, and VGG-16), BNN training procedures, and computational costs. Appendix C presents additional ablation studies, including distributional choices over the simplex, the effect of budget set size, and the effect of the number of closed intervals. It also contains details on fine-tuning large-scale models, hyperparameter settings, ROC and PRC computation procedures, and results on calibration and likelihood evaluation. Appendix D discusses modeling epistemic uncertainty in parameter space versus target space. Appendix E outlines a preliminary idea for extending Epi-Wrapper to regression tasks.

5 Conclusion and Future Directions

This paper introduces *Epistemic Wrapper*, a methodology that extends higher-order uncertainty representation into the parameter space of neural networks. Building on BNNs, it transforms their outputs into belief-function posteriors, enabling richer and more expressive quantification of epistemic uncertainty. The method is robust, efficient, and applicable across architectures, with experiments on four benchmark datasets showing consistent improvements in uncertainty estimation. A limitation is that Epi-Wrapper's performance depends on the quality of the underlying BNN, as poorly trained or high-variance models may yield degraded belief-function outputs. As future work, we aim to integrate Epistemic Wrapper with Bayesian Neural Operators for structured uncertainty quantification in function space, particularly for complex physical systems governed by PDEs. Another direction is to use the wrapped weights to construct predictive random sets in the target space, advancing reliable uncertainty-aware learning.

REFERENCES

Moloud Abdar, Farhad Pourpanah, Sadiq Hussain, Dana Rezazadegan, Li Liu, Mohammad Ghavamzadeh, Paul Fieguth, Xiaochun Cao, Abbas Khosravi, U Rajendra Acharya, et al. A review of uncertainty quantification in deep learning: Techniques, applications and challenges. *Information Fusion*, 76:243–297, 2021.

Gustaf Ahdritz, Aravind Gollakota, Parikshit Gopalan, Charlotte Peale, and Udi Wieder. Provable uncertainty decomposition via higher-order calibration. *arXiv preprint arXiv:2412.18808*, 2024.

David M Blei, Alp Kucukelbir, and Jon D McAuliffe. Variational inference: A review for statisticians. *Journal of the American statistical Association*, 112(518):859–877, 2017.

Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight uncertainty in neural network. In *Proceedings of the International Conference on Machine Learning*, pp. 1613–1622. PMLR, 2015.

Michele Caprio, Maryam Sultana, Eleni Elia, and Fabio Cuzzolin. Credal learning theory. *arXiv preprint arXiv:2402.00957*, 2024.

Matthew Albert Chan, Maria J. Molina, and Christopher Metzler. Estimating epistemic and aleatoric uncertainty with a single model. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=WPxa6OcIdg.

Bertrand Charpentier, Daniel Zügner, and Stephan Günnemann. Posterior network: Uncertainty estimation without ood samples via density-based pseudo-counts. *Advances in neural information processing systems*, 33:1356–1367, 2020.

Fabio Cuzzolin. Geometry of Upper Probabilities. In *ISIPTA*, pp. 188–203, 2003.

Fabio Cuzzolin. Geometry of Dempster's rule of combination. *IEEE Transactions on Systems, Man and Cybernetics part B*, 34(2):961–977, 2004.

Fabio Cuzzolin. Two new Bayesian approximations of belief functions based on convex geometry. *IEEE Transactions on Systems, Man, and Cybernetics - Part B*, 37(4):993–1008, 2007.

Fabio Cuzzolin. On the credal structure of consistent probabilities. In *European Workshop on Logics in Artificial Intelligence*, pp. 126–139. Springer, 2008a.

Fabio Cuzzolin. A geometric approach to the theory of evidence. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 38(4):522–534, 2008b.

Fabio Cuzzolin. Complexes of outer consonant approximations. In *Proceedings of the 10th European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty (ECSQARU'09)*, pp. 275–286, 2009.

Fabio Cuzzolin. Credal semantics of Bayesian transformations in terms of probability intervals. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 40(2):421–432, 2010a.

Fabio Cuzzolin. Geometric conditioning of belief functions. *Proceedings of BELIEF*, 10, 2010b.

Fabio Cuzzolin. The geometry of consonant belief functions: simplicial complexes of necessity measures. *Fuzzy Sets and Systems*, 161(10):1459–1479, 2010c.

Fabio Cuzzolin. On consistent approximations of belief functions in the mass space. In *European Conference on Symbolic and Quantitative Approaches to Reasoning and Uncertainty*, pp. 287–298. Springer, 2011.

Fabio Cuzzolin. Lp consonant approximations of belief functions. *IEEE Transactions on Fuzzy Systems*, 22(2):420–436, April 2014a.

Fabio Cuzzolin. *Belief functions: theory and applications*. Springer, 2014b.

Fabio Cuzzolin. Generalised max entropy classifiers. In Sébastien Destercke, Thierry Denœux, Fabio Cuzzolin, and Arnaud Martin (eds.), *Belief Functions: Theory and Applications*, pp. 39–47, Cham, 2018a. Springer International Publishing.

Fabio Cuzzolin. Visions of a generalized probability theory. *arXiv preprint arXiv:1810.10341*, 2018b.

Fabio Cuzzolin. *The geometry of uncertainty: The geometry of imprecise probabilities*. Springer Nature, 2020.

Fabio Cuzzolin. Uncertainty measures: The big picture. *arXiv preprint arXiv:2104.06839*, 2021.

Fabio Cuzzolin. Uncertainty measures: A critical survey. *Information Fusion*, pp. 102609, 2024.

Fabio Cuzzolin. Lp consonant approximations of belief functions in the mass space. In *Proceedings of the 7th International Symposium on Imprecise Probability: Theory and Applications (ISIPTA'11)*, July 2011.

Fabio Cuzzolin and Ruggero Frezza. Geometric analysis of belief space and conditional subspaces. In *ISIPTA*, pp. 122–132, 2001.

Fabio Cuzzolin and Wenjua Gong. Belief modeling regression for pose estimation. In *Proceedings of the 16th International Conference on Information Fusion*, pp. 1398–1405, 2013.

Arthur P Dempster. Upper and lower probabilities induced by a multivalued mapping. In *Classic works of the Dempster-Shafer theory of belief functions*, pp. 57–72. Springer, 2008.

Stanislav Fort and Stanislaw Jastrzebski. Large scale structure of neural network loss landscapes. *Advances in Neural Information Processing Systems*, 32, 2019.

Dominik Fuchsgruber, Tom Wollschläger, and Stephan Günnemann. Energy-based epistemic uncertainty for graph neural networks. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=6vNPPtWH1Q.

Yarin Gal and Zoubin Ghahramani. Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. In *Proceedings of the International Conference on Machine Learning*, pp. 1050–1059. PMLR, 2016.

Andrew Gelman, John B Carlin, Hal S Stern, David B Dunson, Aki Vehtari, and Donald B Rubin. *Bayesian Data Analysis*. CRC press, 2013.

Wenjuan Gong and Fabio Cuzzolin. A belief-theoretical approach to example-based pose estimation. *IEEE Transactions on Fuzzy Systems*, 26(2):598–611, 2017.

Sebastian G Gruber and Florian Buettner. Uncertainty estimates of predictions via a general bias-variance decomposition. *arXiv preprint arXiv:2210.12256*, 2022.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Timothy Hickey, Qun Ju, and Maarten H Van Emden. Interval arithmetic: From principles to implementation. *Journal of the ACM (JACM)*, 48(5):1038–1068, 2001.

Matthew D Hoffman, Andrew Gelman, et al. The No-U-Turn sampler: Adaptively setting path lengths in Hamiltonian Monte Carlo. *J. Mach. Learn. Res.*, 15(1):1593–1623, 2014.

Paul Hofman, Yusuf Sale, and Eyke Hüllermeier. Quantifying aleatoric and epistemic uncertainty: A credal approach. In *ICML 2024 Workshop on Structured Probabilistic Inference {\&} Generative Modeling*, 2024.

J. R. M. Hosking. L-moments: Analysis and estimation of distributions using linear combinations of order statistics. *Journal of the Royal Statistical Society: Series B (Methodological)*, 52(1):105–124, 12 2018. doi: 10.1111/j.2517-6161.1990.tb01775.x.

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

Laurent Valentin Jospin, Hamid Laga, Farid Boussaid, Wray Buntine, and Mohammed Bennamoun. Hands-on Bayesian neural networks—A tutorial for deep learning users. *IEEE Computational Intelligence Magazine*, 17(2):29–48, 2022.

Mira Juergens, Nis Meinert, Viktor Bengs, Eyke Hüllermeier, and Willem Waegeman. Is epistemic uncertainty faithfully represented by evidential deep learning methods? In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=mxjB0LIgpT.

LS Kim. Understanding the difficulty of training deep feedforward neural networks xavier. In *Proceedings of the International Joint Conference on Neural Networks*, volume 2, 1993.

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

Nikita Kotelevskii, Vladimir Kondratyev, Martin Takáč, Eric Moulines, and Maxim Panov. From risk to uncertainty: Generating predictive uncertainty measures via bayesian estimation. *arXiv preprint arXiv:2402.10727*, 2024.

Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, Technical Report, Computer Science Department, University of Toronto, 2009a.

Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009b. URL http://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf.

Antonis Lambrou, Harris Papadopoulos, and Alex Gammerman. Reliable confidence measures for medical diagnosis with evolutionary algorithms. *IEEE Transactions on Information Technology in Biomedicine*, 15(1):93–99, 2010.

Yann LeCun. The MNIST database of handwritten digits. *http://yann.lecun.com/exdb/mnist/*, 1998.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Isaac Levi. *The enterprise of knowledge: An essay on knowledge, credal probability, and chance*. MIT press, 1980.

Xixun Lin, Wenxiao Zhang, Fengzhao Shi, Chuan Zhou, Lixin Zou, Xiangyu Zhao, Dawei Yin, Shirui Pan, and Yanan Cao. Graph neural stochastic diffusion for estimating uncertainty in node classification. In *Forty-first International Conference on Machine Learning*, 2024.

Jeremiah Liu, Zi Lin, Shreyas Padhy, Dustin Tran, Tania Bedrax Weiss, and Balaji Lakshminarayanan. Simple and principled uncertainty estimation with deterministic deep learning via distance awareness. *Advances in Neural Information Processing Systems*, 33:7498–7512, 2020.

Zhun-Ga Liu, Yu Liu, Jean Dezert, and Fabio Cuzzolin. Evidence combination based on credal belief redistribution for pattern classification. *IEEE Transactions on Fuzzy Systems*, 28(4):618–631, 2019.

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

Shireen Kudukkil Manchingal and Fabio Cuzzolin. Epistemic deep learning. *arXiv preprint arXiv:2206.07609*, 2022.

Shireen Kudukkil Manchingal, Muhammad Mubashar, Kaizheng Wang, Keivan Shariatmadar, and Fabio Cuzzolin. Random-set convolutional neural network (rs-cnn) for epistemic deep learning. *arXiv preprint arXiv:2307.05772*, 2023.

Shireen Kudukkil Manchingal, Muhammad Mubashar, Kaizheng Wang, and Fabio Cuzzolin. A unified evaluation framework for epistemic predictions, 2025a. URL https://arxiv.org/abs/2501.16912.

Shireen Kudukkil Manchingal, Muhammad Mubashar, Kaizheng Wang, Keivan Shariatmadar, and Fabio Cuzzolin. Random-set neural networks. In *The Thirteenth International Conference on Learning Representations*, 2025b. URL https://openreview.net/forum?id=pdjkikvCch.

Dmitry Molchanov, Arsenii Ashukha, and Dmitry Vetrov. Variational dropout sparsifies deep neural networks. In *Proceedings of the International Conference on Machine Learning*, pp. 2498–2507. PMLR, 2017.

Giorgio Morales and John Sheppard. Adaptive sampling to reduce epistemic uncertainty using prediction interval-generation neural networks. *Proceedings of the AAAI Conference on Artificial Intelligence*, Feb. 2025.

Radford M Neal et al. MCMC using Hamiltonian dynamics. *Handbook of Markov Chain Monte Carlo*, 2(11):2, 2011.

Kazuki Osawa, Siddharth Swaroop, Anirudh Jain, Runa Eschenhagen, Richard E. Turner, Rio Yokota, and Mohammad Emtiyaz Khan. Practical deep learning with bayesian principles. *arXiv preprint arXiv:1906.02506*, 2019.

Yusuf Sale, Michele Caprio, and Eyke Höllermeier. Is the volume of a credal set a good measure for epistemic uncertainty? In *Uncertainty in Artificial Intelligence*, pp. 1795–1804. PMLR, 2023.

Murat Sensoy, Lance Kaplan, and Melih Kandemir. Evidential deep learning to quantify classification uncertainty. *Advances in neural information processing systems*, 31, 2018.

Glenn Shafer. *A mathematical theory of evidence*, volume 42. Princeton university press, 1976.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations (ICLR)*, 2015.

Philippe Smets. The transferable belief model and other interpretations of Dempster–Shafer's model. In P. P. Bonissone, M. Henrion, L. N. Kanal, and J. F. Lemmer (eds.), *Uncertainty in Artificial Intelligence*, volume 6, pp. 375–383. North-Holland, Amsterdam, 1991.

Philippe Smets. Belief functions on real numbers. *International Journal of Approximate Reasoning*, 40(3):181–223, 2005.

Andrew Stirn, Tony Jebara, and David A Knowles. A new distribution on the simplex with auto-encoding applications. In *Advances in Neural Information Processing Systems*, 2019. URL https://arxiv.org/abs/1905.12052. arXiv:1905.12052.

Puja Trivedi, Mark Heimann, Rushil Anirudh, Danai Koutra, and Jayaraman J. Thiagarajan. Accurate and scalable estimation of epistemic uncertainty for graph neural networks. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=ZL6yd6N1S2.

Kaizheng Wang, Fabio Cuzzolin, Keivan Shariatmadar, David Moens, and Hans Hallez. Credal wrapper of model averaging for uncertainty estimation on out-of-distribution detection. *arXiv preprint arXiv:2405.15047*, 2024a.

Kaizheng Wang, Fabio Cuzzolin, Keivan Shariatmadar, David Moens, Hans Hallez, et al. Credal deep ensembles for uncertainty quantification. *Advances in Neural Information Processing Systems*, 37: 79540–79572, 2024b.

Kaizheng Wang, Keivan Shariatmadar, Shireen Kudukkil Manchingal, Fabio Cuzzolin, David Moens, and Hans Hallez. Creinns: Credal-set interval neural networks for uncertainty estimation in classification tasks. *Neural Networks*, pp. 107198, 2025.

Larry A. Wasserman. Belief functions and statistical inference. *Canadian Journal of Statistics*, 18(3): 183–196, 1990.

Yeming Wen, Paul Vicol, Jimmy Ba, Dustin Tran, and Roger Grosse. Flipout: Efficient pseudo-independent weight perturbations on mini-batches. In *Proceedings of the International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=rJNpifWAb.

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

A Theoretical Background Concepts

A.1 Belief Functions

Belief functions Cuzzolin (2014b; 2018b), grounded in the mathematical framework of random sets, were initially introduced by Dempster (Dempster, 2008) and later formalized by Shafer (Shafer, 1976) as an alternative model for subjective belief to Bayesian probability. As mathematical objects, they have been extensively studied from an original geometric point of view Cuzzolin & Frezza (2001); Cuzzolin (2003; 2004; 2008b; 2010c;b). Ways of transforming belief functions into Bayesian probabilities or possibility measures have also been investigated Cuzzolin (2009; 2010a; 2011; 2014a; July 2011; 2007).

In finite domains, such as a collection of classes, belief functions are characterised by a *basic probability assignment* (BPA) (Shafer, 1976), which is a set function $m : 2^\Theta \to [0, 1]$ satisfying $m(\emptyset) = 0$ and $\sum_{A\subseteq\Theta} m(A) = 1$. The value $m(A)$ is interpreted as the probability mass directly assigned to subset $A \subseteq \Theta$ in a random-set formulation (Smets, 1991). Subsets A of Θ with $m(A) > 0$ are referred to as *focal elements*. Classical belief functions extend the notion of discrete mass functions by assigning normalized, non-negative mass values not only to elements $\theta \in \Theta$ but to subsets of Θ, governed by:

$$m(A) \geq 0, \forall A \subseteq \Theta, \quad \sum_{A\subseteq\Theta} m(A) = 1. \tag{7}$$

The belief function $Bel(A)$ associated with a mass function m is defined as the total mass assigned to all subsets $B \subseteq A$. Conversely, m can be recovered from Bel through Moebius inversion Shafer (1976):

$$Bel(A) = \sum_{B\subseteq A} m(B), \quad m(A) = \sum_{B\subseteq A} (-1)^{|A\setminus B|} Bel(B). \tag{8}$$

This formulation demonstrates that classical probability measures are a special case of belief functions, assigning mass exclusively to singletons.

A.2 Dirichlet Distribution

The Dirichlet distribution is a continuous multivariate probability distribution defined over the $(K-1)$ simplex:

$$\Delta^{K-1} = \left\{ \boldsymbol{x} \in \mathbb{R}^K \;\middle|\; x_i \geq 0, \sum_{i=1}^{K} x_i = 1 \right\}.$$

It is parameterized by a concentration vector $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_K)$ with each $\alpha_i > 0$, and its probability density function is given by:

$$f(x_1, \ldots, x_K; \alpha_1, \ldots, \alpha_K) = \frac{1}{B(\boldsymbol{\alpha})} \prod_{i=1}^{K} x_i^{\alpha_i - 1}, \tag{9}$$

where $B(\boldsymbol{\alpha})$ is the multivariate Beta function:

$$B(\boldsymbol{\alpha}) = \frac{\prod_{i=1}^{K} \Gamma(\alpha_i)}{\Gamma\left(\sum_{i=1}^{K} \alpha_i\right)}.$$

The shape of the Dirichlet distribution is governed by the values of $\boldsymbol{\alpha}$. Higher values lead to more concentrated distributions around the center of the simplex, while lower values result in more dispersed or sparse distributions. Figure 6 illustrates how different $\boldsymbol{\alpha}$ values affect the distribution over the 2D simplex.

The Dirichlet distribution is widely used in Bayesian statistics, especially for modeling topics in documents and for representing uncertainty Gelman et al. (2013).

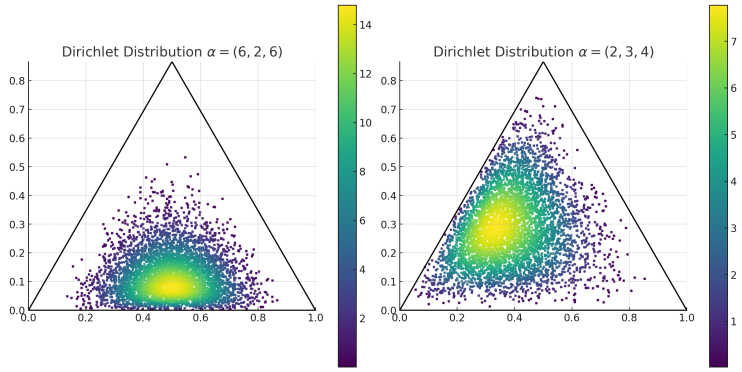


Figure 6: Probability densities of the Dirichlet distribution as functions on the 2D simplex: α = (6,2,6) (left), α = (2,3,4) (right).

A.3 Interval Neural Networks (INNs)

Traditional *interval neural networks* use deterministic interval-based inputs, outputs, and parameters (weights and biases) for each node. The forward propagation in the l^{th} layer of INNs is expressed as:

$$\begin{aligned}[\underline{\boldsymbol{a}}, \overline{\boldsymbol{a}}]^l &= \sigma^l([\underline{\boldsymbol{\omega}}, \overline{\boldsymbol{\omega}}]^l \odot [\underline{\boldsymbol{a}}, \overline{\boldsymbol{a}}]^{l-1} \oplus [\underline{\boldsymbol{b}}, \overline{\boldsymbol{b}}]^l) \\ &= [\sigma^l(\underline{\boldsymbol{o}} + \underline{\boldsymbol{b}}), \sigma^l(\overline{\boldsymbol{o}} + \overline{\boldsymbol{b}})] \text{ with} \\ [\underline{\boldsymbol{o}}, \overline{\boldsymbol{o}}]^l &= [\underline{\boldsymbol{\omega}}, \overline{\boldsymbol{\omega}}]^l \odot [\underline{\boldsymbol{a}}, \overline{\boldsymbol{a}}]^{l-1},\end{aligned} \tag{10}$$

where $\oplus$, $\ominus$, and $\odot$ represent interval addition, subtraction, and multiplication, respectively (Hickey et al., 2001). The terms $[\underline{\boldsymbol{a}}, \overline{\boldsymbol{a}}]^l$, $[\underline{\boldsymbol{a}}, \overline{\boldsymbol{a}}]^{l-1}$, $[\underline{\boldsymbol{\omega}}, \overline{\boldsymbol{\omega}}]^l$, and $[\underline{\boldsymbol{b}}, \overline{\boldsymbol{b}}]^l$ denote the interval-formed outputs of the l^{th} and $(l-1)^{th}$ layers, as well as the intervals of weights and biases of the l^{th} layer, respectively. $\sigma^l(\cdot)$ is the activation function of the l^{th} layer, which must be monotonically increasing. The application of interval arithmetic (Hickey et al., 2001) in eq. equation 10 grants INNs the 'set constraint' property. Specifically, for any $\boldsymbol{a}^{l-1} \in [\underline{\boldsymbol{a}}, \overline{\boldsymbol{a}}]^{l-1}$, $\boldsymbol{\omega}^l \in [\underline{\boldsymbol{\omega}}, \overline{\boldsymbol{\omega}}]^l$, and $\boldsymbol{b}^l \in [\underline{\boldsymbol{b}}, \overline{\boldsymbol{b}}]^l$, the constraint in eq. equation 11 consistently holds.

$$\boldsymbol{a}^l = \sigma^l(\boldsymbol{\omega}^l \cdot \boldsymbol{a}^{l-1} + \boldsymbol{b}^l) \in [\underline{\boldsymbol{a}}, \overline{\boldsymbol{a}}]^l. \tag{11}$$

If $[\underline{\boldsymbol{a}}, \overline{\boldsymbol{a}}]$ is non-negative, such as the output of RELU activation, the calculation of $[\underline{\boldsymbol{o}}, \overline{\boldsymbol{o}}]$ in eq. equation 10 can be simplified as:

$$\begin{aligned}\underline{\boldsymbol{o}} &= \min\{\underline{\boldsymbol{\omega}}, \mathbf{0}\} \cdot \overline{\boldsymbol{a}} + \max\{\underline{\boldsymbol{\omega}}, \mathbf{0}\} \cdot \underline{\boldsymbol{a}} \\ \overline{\boldsymbol{o}} &= \max\{\overline{\boldsymbol{\omega}}, \mathbf{0}\} \cdot \overline{\boldsymbol{a}} + \min\{\overline{\boldsymbol{\omega}}, \mathbf{0}\} \cdot \underline{\boldsymbol{a}}.\end{aligned} \tag{12}$$

B Experimental Setup

Algorithm 1 Epistemic Wrapper Pipeline

Input: Pretrained BNN with posterior samples $\{\mathbf{w}^{(s)}\}$
for each selected weight posterior $\mathbf{w}$ **do**
    Compute mean μ and standard deviation σ
    Truncate posterior using dynamic multiplier
    Discretize intervals into Borel sets and construct belief function Bel
    Compute mass assignments via Möbius inversion
    Normalize mass values and project to 3D simplex
    Compute weighted L-moments L_1, L_2
    Fit Dirichlet distribution using L-moment estimates
end for
Output: Wrapped posteriors represented as Dir($\boldsymbol{\alpha}$)

B.1 Implementation Details

All experiments are implemented using the TensorFlow framework (version 2.13.1), with probabilistic modeling and inference carried out using TensorFlow Probability (TFP) (version 0.21.0). TFP is a library for statistical analysis and probabilistic reasoning that integrates seamlessly with TensorFlow. We employed two Bayesian baseline models: 'BNNR', which uses Auto-Encoding Variational Bayes (Kingma & Welling, 2013) with the local re-parameterization trick (Molchanov et al., 2017), and 'BNNF', which leverages the Flipout gradient estimator along with a negative evidence lower bound (ELBO) loss (Wen et al., 2018). Both baselines are implemented using TFP's built-in variational layers and loss functions. Our complete pipeline, including training, inference, and evaluation of the Epistemic Wrapper is built entirely in TensorFlow 2.13.1 and executed on a machine equipped with 8× NVIDIA A30 GPUs.

B.2 Datasets

We evaluated the performance of the Epistemic Wrapper on four classification benchmarks: MNIST (LeCun, 1998), Fashion-MNIST Xiao et al. (2017), CIFAR-10 (Krizhevsky, 2009a) and CIFAR-100 Krizhevsky (2009b). Following are the details of the iD and OoD datasets.

B.2.1 iD Datasets

MNIST dataset comprises 70,000 greyscale images of handwritten digits (0-9), each with a resolution of 28×28 pixels, and is mostly used for classification and pattern recognition tasks due to its simplicity and accessibility.

Fashion MNIST serves as a more challenging alternative to MNIST, containing 70,000 greyscale images of fashion items, such as shirts, shoes, and bags, also at same resolution of 28×28 pixels. This dataset provides a greater diversity in texture and structure, making it suitable for evaluating model's generalization capabilities.

CIFAR-10 is a collection of 60,000 color images (split into 50,000 training and 10,000 testing samples) across 10 classes, including animals and vehicles, with each image having a resolution of 32×32 pixels. CIFAR-10 is particularly valuable for assessing models in tasks involving color and more complex spatial patterns.

CIFAR-100 consists of 60,000 color images, each of size 32×32 pixels with three RGB channels, divided into 50,000 training images and 10,000 test images. The dataset contains 100 fine-grained classes, with each class having 600 samples, making it a more challenging extension of the CIFAR-10 dataset. Unlike CIFAR-10, which includes only 10 broad categories, CIFAR-100 introduces a hierarchical structure, grouping its 100 classes into 20 superclasses based on semantic similarity.

B.2.2 OoD Datasets

SVHN (Street View House Numbers) is a real-world image dataset obtained from house numbers captured by Google Street View. It consists of over 600,000 digit images (0–9), cropped from street number plates, with each image being 32×32 pixels in RGB format. Unlike the balanced and object-centric nature of CIFAR datasets, SVHN exhibits high variability in illumination, orientation, and background clutter. It is primarily designed for digit recognition tasks but is widely adopted as an OoD dataset when models are trained on natural object datasets such as CIFAR-10 or CIFAR-100, due to its distinct visual domain.

TinyImageNet is a subset of the ImageNet dataset constructed for benchmarking under constrained input dimensions. It contains 200 object classes with 500 training images, 50 validation images, and 50 test images per class, leading to a total of 100,000 images. Each image is downsampled to 64×64 pixels, significantly smaller than the original ImageNet resolution. The dataset retains substantial intra-class variation and fine-grained categories. Due to its larger diversity and semantic distance from CIFAR classes, TinyImageNet serves as a strong OoD benchmark for evaluating model robustness and generalization.

B.3 Bayesian Baselines

In our current implementation, we employ standard and widely-used VI methods such as Auto-Encoding Variational Bayes with local reparameterization BNNR (Auto-Encoding Variational Bayes (Kingma & Welling, 2013) with the local re-parameterization trick (Molchanov et al., 2017)), and BNNF (Flipout gradient estimator with the negative evidence lower bound loss (Wen et al., 2018)). These provide reliable and efficient posterior approximations and are well-supported in TensorFlow, offering a stable foundation for Bayesian neural network training. The core design of the Epistemic Wrapper is independent of the specific VI family used to approximate the posterior. It operates on sampled posterior weights, and thus can naturally extend to richer VI families. We chose mean-field-based VI families in this initial study for their computational tractability and widespread adoption.

B.4 Backbones

We utilized four backbone models in our experiments: MLP, LeNet-5 LeCun et al. (1998), ResNet-18 (He et al., 2016), and VGG-16 Simonyan & Zisserman (2015). The architectural details of each backbone are provided below.

MLP is composed of an input layer, a single hidden layer and an output layer. The input layer processes the input data with a shape that corresponds to the dimensions of the dataset. For grayscale datasets (MNIST and Fashion MNIST), the input shape is $28 \times 28 \times 1$, and for CIFAR-10 and CIFAR-100, the input shape is $32 \times 32 \times 3$. A flattening layer flattens the input into a single-dimensional vector to be fed to the subsequent dense layers. 'DenseFlipout Layers' are implemented using TFP. They approximate the weight posterior distributions using a Flipout Monte Carlo estimator, which reduces the variance of gradient estimates during backpropagation. The first dense layer contains hidden units with ReLU activation, followed by a dropout layer to prevent overfitting. The second dense layer, which acts as the output layer, maps to the number of classes in the dataset.

LeNet-5 architecture is adapted into a fully Bayesian framework using variational inference with Flipout layers from TensorFlow Probability. Both convolutional and fully connected layers are replaced with their Flipout-based counterparts. The input shape is dataset dependent: $28 \times 28 \times 1$ for grayscale datasets such as MNIST and Fashion-MNIST, and $32 \times 32 \times 3$ for RGB datasets such as CIFAR-10 and CIFAR-100. The model begins with a 'Convolution2DFlipout' layer with 6 filters and a 5×5 kernel, followed by an average pooling layer. This is followed by a second 'Convolution2DFlipout' layer with 16 filters and another 5×5 kernel, again followed by average pooling. The output is then flattened and passed through two fully connected variational layers: a 'DenseFlipout' layer with 120 units and ReLU activation, followed by a second 'DenseFlipout' layer with 84 units. The final classification is performed by a 'DenseFlipout' layer with softmax activation and a number of units equal to the number of classes.

ResNet-18 model leverages Bayesian Convolutional Neural Networks (Bayesian CNNs) with Flipout and Reparameterization layers from TensorFlow Probability, enabling weight uncertainty modeling. The architecture consists of four main residual blocks, with convolutional layers followed by batch normalization and ReLU activation. The convolutional layers employ Bayesian weight posterior distributions, where the kernel weights follow a Gaussian posterior parameterized by mean and variance. These distributions are constrained using a log-variance regularization technique, ensuring numerical stability. The weight posteriors are sampled using the Mean-Field Variational Inference approach, enabling Bayesian updates during training. The ResNet-18 backbone begins with an initial convolutional layer followed by four residual blocks, each progressively increasing the number of filters from 64 to 512. The residual connections allow gradient flow through the network, ensuring stable training. The final layers include average pooling, flattening, and a fully connected Bayesian dense layer with Flipout, producing the classification logits.

VGG-16 model integrates Bayesian inference into the classical VGG-16 architecture to enable principled uncertainty estimation in deep learning. The standard convolutional layers are replaced with Bayesian Convolutional Neural Networks (Bayesian CNNs) using Convolution2DReparameterization and Convolution2DFlipout layers from TensorFlow Probability. These layers approximate posterior distributions over weights using Mean-Field Variational Inference, ensuring reliable uncertainty quantification. VGG-16 follows a deep convolutional architecture with 16 layers, consisting of multiple stacked convolutional layers with small 3×3 filters, followed by max pooling layers to progressively reduce spatial dimensions. The Bayesian adaptation maintains this structure while

introducing posterior weight sampling in convolutional layers, ensuring that the feature extraction process incorporates uncertainty information. Batch normalization and ReLU activation are applied to enhance convergence stability, while Bayesian priors constrain weight posteriors, preventing overconfidence in predictions. The final classification layers include Bayesian fully connected layers with Flipout, which sample weights during inference to produce uncertainty-aware predictions.

BNNR vs. BNNF instantiations: Both BNNR and BNNF share the same Bayesian architectures of all above mentioned backbones but differ in their variational inference strategies:

BNNR (Bayesian Neural Network with Reparameterization) uses the Auto-encoding Variational Bayes Kingma & Welling (2013) along with the local reparameterization trick Molchanov et al. (2017). This variant estimates the evidence lower bound (ELBO) and applies Gaussian posterior sampling locally per activation.

BNNF (Bayesian Neural Network with Flipout) uses the Flipout estimator Wen et al. (2018), which decorrelates the gradient estimates across examples in a mini-batch, leading to lower variance during optimization. BNNF directly applies Flipout-based sampling in both convolutional and dense layers and uses a negative ELBO as the loss function.

These two baselines allow us to evaluate the effectiveness of our Epistemic Wrapper under different stochastic inference regimes.

B.5 Training Details

As a baseline, we use standard variational Bayesian Neural Networks (BNNs) (Blei et al., 2017), starting with a classical Multilayer Perceptron (MLP) architecture. The Bayesian MLP is trained using the Evidence Lower Bound (ELBO) objective, which combines the negative log-likelihood (NLL) with a Kullback-Leibler (KL) divergence regularization term. The NLL is computed via softmax cross-entropy, and the KL divergence measures the distance between the approximate posterior and the prior distributions over the weights. The MLP models are trained for 20 epochs on MNIST and Fashion-MNIST using the Adam optimizer. The Bayesian LeNet-5 is trained under both BNNR and BNNF setups using the same ELBO-based training strategy. Models are trained for 500 epochs on CIFAR-10 with Adam optimizer, batch size 32. We apply standard data augmentation techniques (random horizontal flipping and cropping). For deeper architectures such ResNet-18 and VGG-16, we adopt dataset-specific training schedules. Models trained on CIFAR-10 are trained for 50 epochs, while those trained on CIFAR-100 are trained for 200 epochs. Batch normalization is applied after each convolutional layer. Both BNNR and BNNF variants are implemented using Flipout-compatible variational layers from TensorFlow Probability and are trained from scratch on a single NVIDIA A30 GPU.

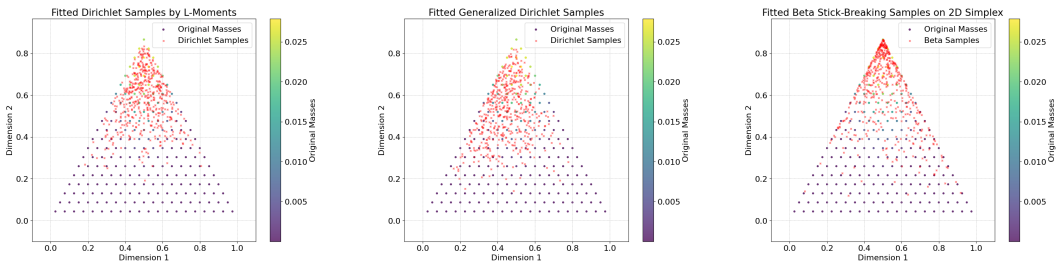
B.6 Computational Cost

The computational overhead introduced by the Epistemic Wrapper is manageable. Because the wrapper operates on a small, selected subset of network weights 5% in MLPs and only 0.1% in large-scale models based on a posterior selection criterion. Importantly, it is applied post hoc to sampled weights and does not interfere with the training process of the underlying Bayesian Neural Network. As such, the forward and backward passes remain unaffected. All experiments were conducted on a machine equipped with 8× NVIDIA A30 GPUs, with each run executed on a single GPU. The added runtime during inference and evaluation phases was minor and remained well within the limits of practical deployment. This indicates that the Epistemic Wrapper retains the scalability and efficiency of the underlying BNNs while providing improved uncertainty estimation. Our method is compatible with existing BNN pipelines and does not add a significant computational burden, making it suitable for large-scale or resource-constrained applications.

C Experimental Results

C.1 Additional Ablation Studies

Distributional Choices over the Simplex: To assess the modelling choice of using a Dirichlet distribution for belief posteriors representation, we perform an ablation study comparing it to two



(a) Dirichlet fitted using L-MoM (b) Generalized Dirichlet (c) Beta Distribution

Figure 7: Distributional Choices over the Simplex

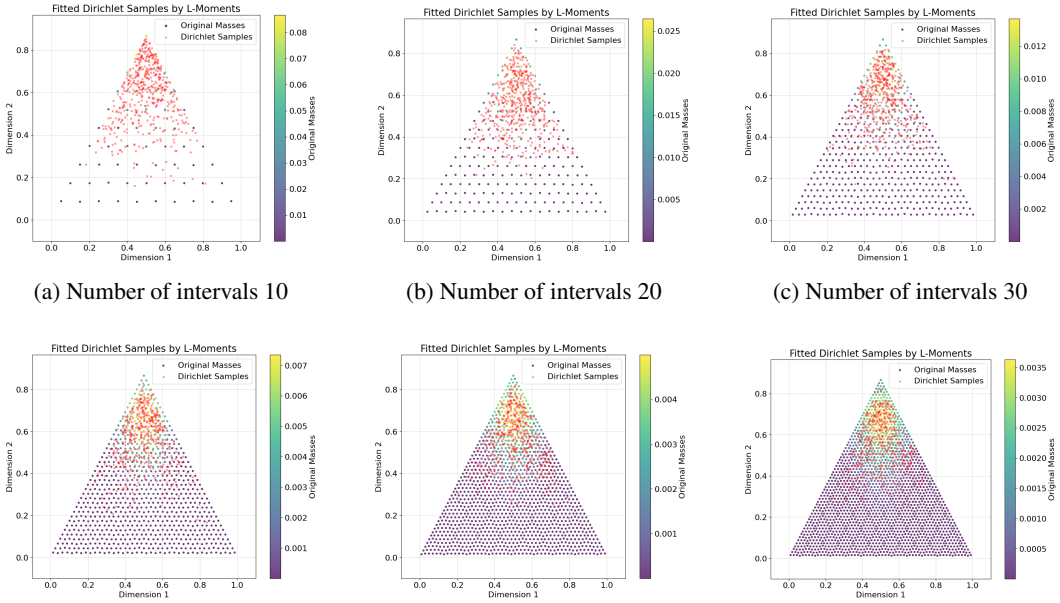
alternative probability distributions defined over the simplex. The following experiments compare the Dirichlet with a Generalized Dirichlet and a Beta stick-breaking model, evaluating their ability to capture the structure of the normalized masses in the proposed methodology. While any valid Probability Density Function (PDF) defined over the simplex could, in principle, be used to model belief distributions, Dirichlet distributions have consistently demonstrated empirical effectiveness for representing epistemic uncertainty in neural networks Stirn et al. (2019). To assess this modelling choice more critically, we conduct an ablation study comparing the use of three distributions defined on the probability simplex: (i) a standard Dirichlet distribution fitted using the method of L-moments, (ii) a Generalized Dirichlet distribution, and (iii) a Beta stick-breaking model. The Dirichlet distribution is used as our baseline because it offers a mathematically simple, symmetric, and interpretable formulation that is widely adopted in evidential deep learning literature. It allows us to encode a single mode of belief mass and is straightforward to parameterize using closed-form moments. The Generalized Dirichlet distribution extends this by introducing additional flexibility via a second set of parameters, enabling skewness and more complex belief shapes. Finally, the Beta stick-breaking model introduces an alternative constructive approach by allocating belief mass sequentially from Beta-distributed proportions, resulting in a valid but directionally expressive distribution over the simplex Stirn et al. (2019). All three models are fitted to the same belief mass vectors, and their sample distributions are visualized in Figures 7a, 7b, and 7c. While the Dirichlet and Generalized Dirichlet distributions align well with the center of mass observed in the belief functions, the Beta stick-breaking model demonstrates an equally valid fit but with a distinct shape and construction. This ablation supports the rationale that, while various PDFs on the simplex are valid for representing belief functions, the Dirichlet distribution remains a principled choice due to its mathematical simplicity, interpretability and established use in modelling epistemic uncertainty in neural networks.

Effect of Budgeting Set Size: Table 4 presents the classification accuracy of the Epistemic Wrapper on MNIST across varying budgeting percentages, both before and after fine-tuning. The percentage indicates the fraction of posterior weights selected for wrapping based on a high-mean criterion, while the number of intervals was fixed at 30. Before fine-tuning, we observe that moderate budgeting (e.g., 10% or 20%) results in significantly improved performance over the baseline INN, which achieved only 9.33% accuracy. In particular, wrapping only 10% of the weights led to a substantial increase in accuracy (45.34%), indicating that even a small subset of informative weights contributes meaningfully to uncertainty-aware decision-making. However, performance declines when budgeting exceeds 20%, which may be attributed to the limited capacity of the small-scale MLP model (with only 8 hidden units). In such a constrained architecture, wrapping a larger fraction of weights may reduce generalization by overfitting or introducing excessive variance in the wrapped ensemble.

After fine-tuning, performance improves and stabilizes across all budgeting levels. The accuracy remains consistently above 91% even with minimal weight wrapping, indicating that fine-tuning effectively adjusts the selected wrapped weights to better align with the underlying predictive task. The best performance (91.85%) is achieved at 30% budgeting, but all configurations from 10% to 50% perform comparably well, highlighting the robustness of the approach after refinement. These results confirm that wrapping a small subset of epistemically informative weights can significantly enhance predictive performance.

Table 4: Performance comparison on MNIST across different budgeting percentages before and after fine-tuning.

STRATEGY	INN ACCURACY (%)	EPI-WRAPPER BUDGETING PERCENTAGE (SELECTED WEIGHTS)					
		5%	10%	20%	30%	40%	50%
BEFORE FINE-TUNING	9.33 ± 0.54	25.46 ± 1.57	$\mathbf{45.34 \pm 1.38}$	42.25 ± 1.26	32.62 ± 2.00	19.07 ± 0.80	14.08 ± 0.94
AFTER FINE-TUNING	91.12 ± 0.08	91.08 ± 0.09	91.83 ± 0.04	91.84 ± 0.04	$\mathbf{91.85 \pm 0.13}$	91.82 ± 0.09	91.50 ± 0.05



(a) Number of intervals 10 (b) Number of intervals 20 (c) Number of intervals 30

(d) Number of intervals 40 (e) Number of intervals 50 (f) Number of intervals 60

Figure 8: Varying Number of closed intervals

Effect of Number of Closed Intervals: To assess the impact of closed intervals for computing belief values leading to fitting a Dirichlet distribution over grid of mass values (as explained in sections 3.3 and 3.4). We conducted an ablation study by varying the number of intervals before fitting the Dirichlet distribution. Table 5 and Fig. 8 show the resulting α values computed using the method of *L-moments* for different numbers of intervals, with a fixed sample size of 5000. We observe that the estimated parameters stabilize around 30 intervals, beyond which changes become marginal. Based on this observation, we fix the number of intervals to 30 for all subsequent experiments to balance estimation stability and computational efficiency.

All ablation experiments are conducted on the MNIST dataset using a BNN MLP (hidden units= 8, samples =5000).

Table 5: Ablation study of Dirichlet α estimates using method of L-moment across varying numbers of intervals. Results are computed on 5000 samples.

Number of Intervals	**Estimated α Values**
10	[1.0657, 4.5643, 1.0626]
20	[1.5958, 5.7176, 1.4097]
30	[1.6651, 6.1145, 1.6197]
40	[1.5272, 6.2213, 1.5665]
50	[1.6660, 6.3547, 1.6410]
60	[1.6935, 6.4927, 1.6669]

C.2 Fine-tuning on Large-scale Models

For the inference using INNs we have two types of weights (as explained in section 3.5): 'unwrapped' weights, directly sampled from Gaussian posteriors, and 'wrapped' weights that are Dirichlet-derived intervals. Specifically, for a given unwrapped weight with mean μ and standard deviation σ, we define the interval as:

$$\text{Lower Bound} = \mu - k \cdot \sigma, \qquad \text{Upper Bound} = \mu + k \cdot \sigma,$$

where k is a learnable, layer-specific scaling factor.

This scaling factor k serves a similar purpose to the *tempering parameter* τ in variational Bayesian inference Osawa et al. (2019), which modulates the influence of the likelihood in the posterior. Smaller values of k lead to narrower (sharper) intervals, corresponding to more confident parameter estimates; larger values encourage wider bounds and more conservative uncertainty. Unlike fixed-scale approaches (e.g., $\mu \pm \sigma$), our model learns k jointly with the rest of the parameters during training. For numerical stability and positivity, we parameterize k via a softplus transformation: $k = \log(1 + \exp(k_{\text{raw}}))$, where k_{raw} is a trainable tensor. This uncertainty calibration mechanism allows each layer to adaptively control the initial spread of its weights, improving both optimization stability and predictive robustness. This is especially important in deep architectures, where poorly controlled interval propagation may lead to unstable gradients or diluted information. At inference time, our method uses these calibrated intervals to propagate uncertainty. In contrast, the baseline INN applies standard random initialization Kim (1993). Both models undergo fine-tuning, but only the Epi-Wrapper benefits from interval-aware initialization based on transformed posteriors.

C.3 Hyperparameter Settings

The experimental setup of the main manuscript contains fixed hyperparameters such as number of closed intervals (30) and samples drawn from posterior distributions (5000). The budgeting strategy is applied consistently across experiments, with 5% of weights selected for small-scale model such as MLP, and 0.1% for larger architectures including LeNet-5, ResNet-18, and VGG-16.

C.4 ROC and PRC Computation for OoD Detection

Scores and Labels. Given in-distribution (ID) samples with scores $\{s_i^{\text{id}}\}_{i=1}^{n_{\text{id}}}$ and out-of-distribution (OoD) samples with scores $\{s_j^{\text{ood}}\}_{j=1}^{n_{\text{ood}}}$, we form

$$\mathbf{s} = \big[s_1^{\text{id}}, \ldots, s_{n_{\text{id}}}^{\text{id}},\ s_1^{\text{ood}}, \ldots, s_{n_{\text{ood}}}^{\text{ood}}\big], \qquad \mathbf{y} = \big[0, \ldots, 0,\ 1, \ldots, 1\big],$$

where $y = 1$ denotes OoD and $y = 0$ denotes ID. In your code, each score is the predictive entropy

$$H(p) \ = \ -\sum_{c=1}^{C} p_c \log p_c,$$

computed from midpoint logits (averaging two heads) followed by softmax.

ROC: For a threshold τ, predict $\hat{y} = \mathbb{I}[s \geq \tau]$. Define

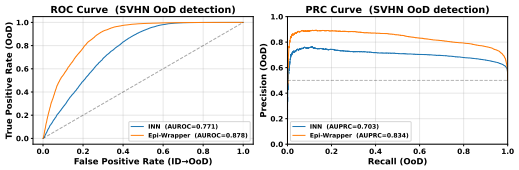
$$\text{TPR}(\tau) = \frac{\text{TP}(\tau)}{\text{TP}(\tau) + \text{FN}(\tau)}, \qquad \text{FPR}(\tau) = \frac{\text{FP}(\tau)}{\text{FP}(\tau) + \text{TN}(\tau)}.$$

Sweeping τ from $+\infty$ to $-\infty$ traces the ROC curve $\big(\text{FPR}(\tau), \text{TPR}(\tau)\big)$. The area under the ROC (AUROC) is computed by the trapezoidal rule over the curve.

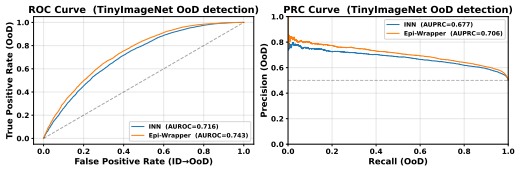
PRC: Precision–Recall uses

$$\text{Precision}(\tau) = \frac{\text{TP}(\tau)}{\text{TP}(\tau) + \text{FP}(\tau)}, \qquad \text{Recall}(\tau) = \text{TPR}(\tau).$$

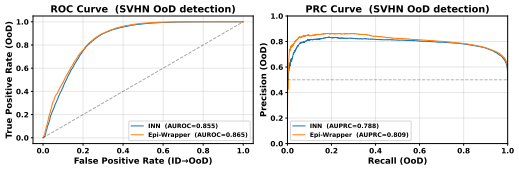
Sweeping τ yields the PRC $\big(\text{Recall}(\tau), \text{Precision}(\tau)\big)$. The area under the PRC (AUPRC) can be computed via the average-precision estimator or trapezoidal integration. A reference baseline is the positive prior $\pi = \frac{n_{\text{ood}}}{n_{\text{id}} + n_{\text{ood}}}$.



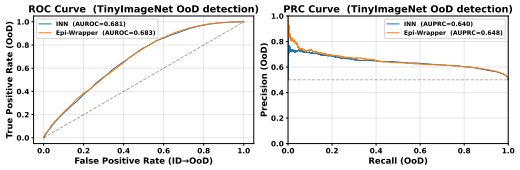
(a) ROC and PRC curves for iD CIFAR-10 vs. SVHN OoD detection using ResNet-18 with BNNR (Molchanov et al., 2017).



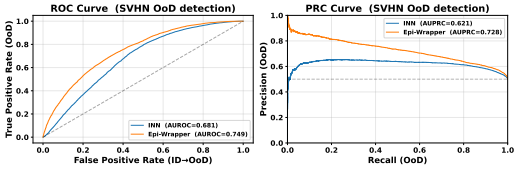
(b) ROC and PRC curves for iD CIFAR-100 vs. TinyImageNet OoD detection using ResNet-18 with BNNR (Molchanov et al., 2017).



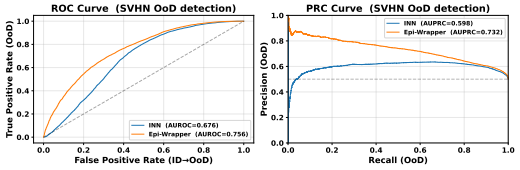
(c) ROC and PRC curves for iD CIFAR-10 vs. SVHN OoD detection using VGG-16 with BNNR (Molchanov et al., 2017).



(d) ROC and PRC curves for iD CIFAR-100 vs. TinyImageNet OoD detection using VGG-16 with BNNF (Wen et al., 2018).



(e) ROC and PRC curves for iD CIFAR-10 vs. SVHN OoD detection using LeNet-5 with BNNR (Molchanov et al., 2017).



(f) ROC and PRC curves for iD CIFAR-100 vs. TinyImageNet OoD detection using LeNet-5 with BNNF (Wen et al., 2018).

Figure 9: ROC and PRC results for OoD detection on SVHN dataset using INN and Epi-Wrapper. The curves illustrate performance trade-offs under entropy-based scoring, with AUROC and AUPRC values reported in the legends.

Comparative Analysis. The ROC and PRC plots in Figure 9 further highlight the performance differences between INN and Epi-Wrapper on OoD detection with SVHN as the outlier dataset. Across both backbone settings (ResNet-18 with BNNF and VGG-16 with BNNR), the Epi-Wrapper curves consistently dominate the INN baseline, reflecting higher true positive rates at lower false positive rates in ROC space, as well as improved precision across recall levels in PRC space. These visual trends align with the quantitative results reported in Table 3, confirming that parameter-space uncertainty modeling via Epi-Wrapper yields superior separability between in-distribution and out-of-distribution samples. The improvement is particularly pronounced in the PRC, where Epi-Wrapper maintains high precision even at challenging recall levels, suggesting its robustness under class-imbalance conditions common in OoD detection tasks.

C.5 Calibration and Likelihood Evaluation

To evaluate the probabilistic calibration of our models, we compute two key metrics: 'Expected Calibration Error (ECE)' and 'Negative Log Likelihood (NLL)'. These metrics assess the alignment between predicted confidence and actual correctness (ECE), and the quality of probabilistic predictions (NLL).

Expected Calibration Error (ECE): quantifies the average discrepancy between the predicted confidence of a model and the observed precision. Predictions are binned into M intervals (we use $M = 15$). For each bin, the average confidence and accuracy are computed, where $\text{conf}(B_m)$ is the average predicted maximum probability of samples in bin B_m. The ECE is the weighted average of the absolute difference between these values across all bins:

Table 6: ECE and NLL for INN (baseline) vs. Epi-Wrapper (ours) across MNIST, Fashion-MNIST, CIFAR-10, and CIFAR-100 datasets using BNNF and BNNR baselines with MLP, LeNet-5, ResNet-18, and VGG-16 backbones. (Classification accuracy results are reported separately in Table 2.)

Dataset	Backbone	# Params	Baseline	ECE (↓)		NLL (↓)	
				INN	Epi-Wrapper	INN	Epi-Wrapper
MNIST	MLP	12.7K	BNNF	0.011 ± 0.031	$\mathbf{0.009 \pm 0.018}$	0.388 ± 0.007	$\mathbf{0.298 \pm 0.004}$
Fashion MNIST	MLP	12.7K	BNNF	0.035 ± 0.012	$\mathbf{0.006 \pm 0.090}$	0.410 ± 0.018	$\mathbf{0.332 \pm 0.019}$
CIFAR-10	LeNet-5	166.3K	BNNF	0.060 ± 0.061	$\mathbf{0.059 \pm 0.019}$	1.365 ± 0.090	$\mathbf{1.210 \pm 0.045}$
			BNNR	0.060 ± 0.061	$\mathbf{0.053 \pm 0.006}$	1.365 ± 0.090	$\mathbf{1.608 \pm 0.009}$
	ResNet-18	9.82M	BNNF	0.084 ± 0.015	$\mathbf{0.041 \pm 0.013}$	0.616 ± 0.070	$\mathbf{0.333 \pm 0.012}$
			BNNR	0.084 ± 0.015	$\mathbf{0.053 \pm 0.012}$	0.616 ± 0.070	$\mathbf{0.422 \pm 0.057}$
	VGG-16	30.24M	BNNF	0.073 ± 0.012	$\mathbf{0.069 \pm 0.087}$	0.520 ± 0.081	$\mathbf{0.451 \pm 0.033}$
			BNNR	0.073 ± 0.012	$\mathbf{0.071 \pm 0.010}$	0.520 ± 0.081	$\mathbf{0.462 \pm 0.054}$
CIFAR-100	ResNet-18	9.8M	BNNF	0.059 ± 0.049	$\mathbf{0.055 \pm 0.087}$	1.460 ± 0.084	$\mathbf{1.451 \pm 0.088}$
			BNNR	0.059 ± 0.049	$\mathbf{0.051 \pm 0.056}$	1.460 ± 0.084	$\mathbf{1.437 \pm 0.010}$
	VGG-16	30.24M	BNNF	0.043 ± 0.023	$\mathbf{0.040 \pm 0.013}$	2.141 ± 0.089	$\mathbf{1.999 \pm 0.071}$
			BNNR	0.043 ± 0.023	$\mathbf{0.039 \pm 0.045}$	2.141 ± 0.089	$\mathbf{1.766 \pm 0.059}$

$$\text{ECE} = \sum_{m=1}^{M} \frac{|B_m|}{n} \left|\text{acc}(B_m) - \text{conf}(B_m)\right|, \tag{13}$$

where n is the total number of samples, and B_m denotes the m-th bin.

Negative Log Likelihood (NLL): captures how well a model's predicted probabilities match the true labels. It penalizes incorrect predictions with high confidence and rewards well-calibrated probability distributions:

$$\text{NLL} = -\frac{1}{n} \sum_{i=1}^{n} \log \hat{p}_{i,y_i}, \tag{14}$$

where $\hat{p}_{i,y_i}$ is the predicted probability for the correct class.

Table 6 summarizes the performance of the baseline INN and the proposed Epi-Wrapper model on the four datasets. Across almost all settings, Epi-Wrapper achieves lower ECE and NLL compared to the INN baseline, highlighting its superior probabilistic calibration and confidence estimation. These improvements suggest that, beyond classification accuracy, Epi-Wrapper provides more trustworthy uncertainty estimates, which is especially important for decision-critical applications where model confidence is as crucial as prediction correctness.

D Modeling Epistemic Uncertainty: Parameter vs. Target Space

Epistemic uncertainty (EU) originates from limited knowledge about the model parameters themselves, a view grounded in Bayesian learning where posterior distributions over weights directly encode parameter uncertainty. Although several recent approaches such as Evidential Deep Learning (EDL) (Sensoy et al., 2018) and credal set-based classification (Wang et al., 2024b) have focused on modeling uncertainty in the target (output) space, these methods do not explicitly represent uncertainty over model parameters. In contrast, our Epistemic Wrapper framework introduces a second-order uncertainty representation in the parameter space via belief functions. By wrapping posterior distributions derived from BNNs, our approach captures variability before the prediction stage, allowing for richer epistemic modeling. This can be particularly beneficial in scenarios with limited training data or when encountering out-of-distribution (OoD) inputs.

Approaches that operate solely in the output space such as decompositions of predictive uncertainty via proper scoring rules, Bregman divergences, calibration methods, or generalized bias-variance analyses (Kotelevskii et al., 2024; Ahdritz et al., 2024; Gruber & Buettner, 2022) provide important insights into the behaviour of predictive distributions. However, these techniques characterize EU only indirectly, after uncertainty has already propagated through the model. In contrast, our framework addresses uncertainty at its source by representing and enriching parameter-space posteriors before

they influence predictions. This distinction allows us to capture model-level uncertainty in a principled way, offering a complementary perspective to output-based decompositions. By integrating belief functions and Dirichlet representations directly in parameter space, our method provides a prior-agnostic and interpretable mechanism for epistemic uncertainty quantification, while remaining fully compatible with existing Bayesian neural networks.

We stress that parameter-space and target-space approaches are not substitutes but complementary perspectives. Target-space methods characterize predictive uncertainty directly, while parameter-space methods quantify the model's internal epistemic state before it influences predictions. Our experiments show that Epistemic Wrapper consistently improves on standard downstream tasks such as OoD detection, calibration (ECE), and likelihood evaluation (NLL), providing strong empirical evidence that parameter-space epistemic modeling yields practical benefits in addition to its theoretical grounding. In summary, Epistemic Wrapper serves as a principled complement to target-based methods, enriching parameter-space uncertainty modeling. Importantly, our method is not directly comparable to target-based approaches such as EDL or ensembles, because they operate in a fundamentally different space.

E Epi-Wrapper for Regression Tasks

In principle, the Epistemic Wrapper is not restricted to classification tasks. The proposed framework models epistemic uncertainty in the parameter space of Bayesian Neural Networks via belief functions, which is independent of the nature of the output space. The only requirement for extending the method to regression tasks is the availability of a suitable Interval Neural Network (INN) or similar regression architecture capable of consuming interval-valued parameters during inference. While we have focused on classification tasks in this work, we believe that extending the approach to regression is a natural and promising direction for future work.