Posterior Network: Uncertainty Estimation without OOD Samples via Density-Based Pseudo-Counts

Bertrand Charpentier, Daniel Zügner, Stephan Günnemann
Technical University of Munich, Germany
{charpent, zuegnerd, guennemann}@in.tum.de

Abstract

Accurate estimation of aleatoric and epistemic uncertainty is crucial to build safe and reliable systems. Traditional approaches, such as dropout and ensemble methods, estimate uncertainty by sampling probability predictions from different submodels, which leads to slow uncertainty estimation at inference time. Recent works address this drawback by directly predicting parameters of prior distributions over the probability predictions with a neural network. While this approach has demonstrated accurate uncertainty estimation, it requires defining arbitrary target parameters for in-distribution data and makes the unrealistic assumption that out-of-distribution (OOD) data is known at training time.

In this work we propose the Posterior Network (PostNet), which uses Normalizing Flows to predict an individual closed-form posterior distribution over predicted probabilites for any input sample. The posterior distributions learned by PostNet accurately reflect uncertainty for in- and out-of-distribution data – without requiring access to OOD data at training time. PostNet achieves state-of-the art results in OOD detection and in uncertainty calibration under dataset shifts.

1 Introduction

Quantifying uncertainty in neural network predictions is key to making Machine Learning reliable. In many sensitive domains, like robotics, financial or medical areas, giving autonomy to AI systems is highly dependent on the trust we can assign to them. In addition, AI systems being aware about their predictions' uncertainty, can adapt to new situations and refrain from taking decisions in unknown or unsafe conditions. Despite of this necessity, traditional neural networks show overconfident predictions, even for data that is signifanctly different from the training data [21] [11]. In particular, they should distinguish between *aleatoric* and *epistemic* uncertainty, also called data and knowledge uncertainty [24], respectively. The aleatoric uncertainty is irreducible from the data, e.g. a fair coin has 50/50 chance for head. The epistemic uncertainty is due to the lack of knowledge about unseen data, e.g. an image of an unknown object or an outlier in the data.

Related work. Uncertainty estimation is a growing research area unifying various approaches [3, 23, 30, 7, 21, 8]. Bayesian Neural Networks learn a *distribution* over the weights [3, 23, 30]. Another class of approaches uses a collection of sub-models and aggregates their predictions, which are in turn used to estimate statistics (e.g., mean and variance) of the class probability distribution. Even though such methods, like ensemble and drop-out, have demonstrated remarkable performance (see, e.g., [38]), they describe implicit distributions for predictions and require a costly sampling phase at inference time for uncertainty estimation.

Recently, a new class of models aims to directly predict the parameters of a prior distribution on the categorical probability predictions, accounting for the different types of uncertainty [24, 25, 35, 1]. However, these methods require (i) the definition of arbitrary target prior distributions [24, 25, 35],
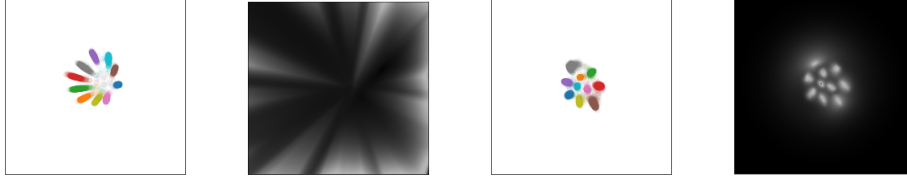
(a) Data labels - PriorNet (b) Uncertainty - PriorNet (c) Data labels - PostNet (d) Uncertainty - PostNet

Figure 1: PriorNet has a pre-ultimate layer of dimension 2 and was trained with Reverse KL and uniform noise on $[0, 255]^{28 \times 28}$ as OOD data. PostNet has a latent space of dimension 2 and was trained without OOD data. (a) and (c) show the learned latent positions of data with colored labels. (b) and (d) show uncertainty estimates in the latent spaces where darker regions indicate high uncertainty. PostNet correctly assigns high uncertainty to OOD regions contrary to PriorNet.

and most importantly, (ii) out-of-distribution (OOD) samples during training time, which is an unrealistic assumption in most applications [24, 25].

Classically, these models would use both ID and OOD samples (e.g. MNIST and FashionMNIST) during training to detect similar OOD samples (i.e. FashionMNIST) at inference time. We show that preventing access to explicit OOD data during training leads to poor results using these approaches (see Fig. 1 for MNIST; or app. for toy datasets). Contrary to the expected results, these models produce increasingly confident predictions for samples far from observed data. In contrast, we propose Posterior Network (PostNet), which assigns high epistemic uncertainty to out-of-distribution samples, low overall uncertainty to regions nearby observed data of a single class, and high aleatoric and low epistemic uncertainty to regions nearby observed data of different classes.

PostNet uses normalizing flows to learn a distribution over Dirichlet parameters in latent space. We enforce the densities of the individual classes to integrate to the number of training samples in that class, which matches well with the intuition of Dirichlet parameters corresponding to the number of observations per class. PostNet does not require any OOD samples for training, the (arbitrary) specification of target prior distributions, or costly sampling for uncertainty estimation at test time.

2 Posterior Network

In classification, we can distinguish between two types of uncertainty for a given input $\mathbf{x}^{(i)}$: the uncertainty on the class prediction $y^{(i)} \in \{1, \ldots, C\}$ (i.e. aleatoric unceratinty), and the uncertainty on the categorical distribution prediction $\mathbf{p}^{(i)} = [p_1^{(i)}, \ldots, p_C^{(i)}]$ (i.e. epistemic uncertainty). A convenient way to model both is to describe the *epistemic distribution* $q^{(i)}$ of the categorical distribution prediction $\mathbf{p}^{(i)}$, i.e. $\mathbf{p}^{(i)} \sim q^{(i)}$. From the epistemic distribution follows naturally an estimate of the *aleatoric distribution* of the class prediction $y^{(i)} \sim \text{Cat}(\bar{\mathbf{p}}^{(i)})$ where $\mathbb{E}_{q^{(i)}}[\mathbf{p}^{(i)}] = \bar{\mathbf{p}}^{(i)}$.

Approaches like ensembles [21] and dropout [8] model $q^{(i)}$ implicitly, which only allows them to estimate statistics at the cost of S samples (e.g. $\mathbb{E}_{q^{(i)}}[\mathbf{p}^{(i)}] \approx \frac{1}{S} \sum_{s=1}^{S} \tilde{\mathbf{p}}^{(i)}$ where $\tilde{\mathbf{p}}^{(i)}$ is sampled from $q^{(i)}$). Another class of models [24, 25, 1, 35] explicitly parametrizes the epistemic distribution with a Dirichlet distribution (i.e. $q^{(i)} = \text{Dir}(\boldsymbol{\alpha}^{(i)})$ where $f_\theta(x^{(i)}) = \boldsymbol{\alpha}^{(i)} \in \mathbb{R}_+^C$), which is the natural prior for categorical distributions. This parametrization is convenient since it requires only one pass to compute epistemic distribution, aleatoric distribution and class prediction:

$$q^{(i)} = \text{Dir}(\boldsymbol{\alpha}^{(i)}), \qquad \bar{p}_c^{(i)} = \frac{\alpha_c}{\alpha_0} \text{ with } \alpha_0 = \sum_{c=1}^{C} \alpha_c, \qquad y^{(i)} = \arg\max [\bar{p}_1, ..., \bar{p}_C] \qquad (1)$$

The concentration parameters $\alpha_c^{(i)}$ can be interpreted as the number of observed samples of class c and, thus, are a good indicator of epistemic uncertainty for non-degenerate Dirichlet distributions (i.e. $\alpha_c^{(i)} \geq 1$). To learn these parameters, Prior Networks [24, 25] use OOD samples for training and define different target values for ID and OOD data. For ID data, $\alpha_c^{(i)}$ is set to an arbitrary, large number if c is the correct class and 1 otherwise. For OOD data, $\alpha_c^{(i)}$ is set to 1 for all classes.

This approach has four issues: **(1)** The knowledge of OOD data for training is unrealistic. In practice, we might not have these data, since OOD samples are by definition not likely to be observed. **(2)** Discriminating in- from out-of-distribution data by providing an explicit set of OOD samples

is hopeless. Since any data not from the data distribution is OOD, it is therefore impossible to characterize the infinitely large OOD distribution with an explicit data set. **(3)** The predicted Dirichlet parameters can take any value, especially for new OOD samples which were not seen during training. In the same way, the sum of the total fictitious prior observations over the full input domain $\int \alpha_0(\mathbf{x})d\mathbf{x}$ is not bounded and in particular can be much larger than the number of ground-truth observations N. This can result in undesired behavior and assign arbitrarily high epistemic certainty for OOD data not seen during training. **(4)** Besides producing such arbitrarily high prior confidence, PNs can also produce degenerate concentration parameters (i.e. $\alpha_c < 1$). While [25] tried to fix this issue by using a different loss, nothing intrinsically prevents Prior Networks from predicting degenerate prior distributions. In the following section we describe how Posterior Network solves these drawbacks.

2.1 An input-dependent Bayesian posterior

First, recall the Bayesian update of a single categorical distribution $y \sim \text{Cat}(\mathbf{p})$. It consists in (1) introducing a prior Dirichlet distribution over its parameters i.e. $\mathbb{P}(\mathbf{p}) = \text{Dir}(\boldsymbol{\beta}^{\text{prior}})$ where $\boldsymbol{\beta}^{\text{prior}} \in \mathbb{R}^C_+$, and (2) using N given observations $y^{(1)}, ..., y^{(N)}$ to form the posterior distribution $\mathbb{P}(\mathbf{p}|\{y^{(j)}\}_{j=1}^N) = \text{Dir}(\boldsymbol{\beta}^{\text{prior}} + \boldsymbol{\beta}^{\text{data}})$ where $\beta_c^{\text{data}} = \sum_j \mathbb{1}_{y^{(j)}=c}$ are the class counts. That is, the Bayesian update is

$$\mathbb{P}(\mathbf{p}|\{y^{(j)}\}_{j=1}^n) \propto \mathbb{P}(\{y^{(j)}\}_{j=1}^n|\mathbf{p}) \times \mathbb{P}(\mathbf{p}). \tag{2}$$

Observing no data (i.e. $\beta_c^{\text{data}} \rightarrow 0$) would lead to flat categorical distribution (i.e. $p_c = \beta_c^{\text{prior}} \cdot (\sum_{c'} \beta_{c'}^{\text{prior}})^{-1}$), while observing many samples (i.e. β_c^{data} is large) would converge to the true data distribution (i.e. $p_c \approx \frac{\beta_c}{\sum_i \beta_i}$). Furthermore, we remark that N behaves like a certainty budget distributed over all classes i.e. $N = \sum_c \beta_c^{\text{data}}$.

Classification is more complex. Generally, we predict the class label $y^{(i)}$ from a different categorical distribution $\text{Cat}(\mathbf{p}^{(i)})$ for each input $\mathbf{x}^{(i)}$. PostNet extends the Bayesian treatment of a single categorical distribution to classification by predicting an individual posterior update for any possible input. To this end, it distinguishes between a fixed prior parameter $\boldsymbol{\beta}^{\text{prior}}$ and the additional learned (pseudo) counts $\boldsymbol{\beta}^{(i)}$ to form the parameters of the posterior Dirichlet distribution $\boldsymbol{\alpha}^{(i)} = \boldsymbol{\beta}^{\text{prior}} + \boldsymbol{\beta}^{(i)}$. Hence, PostNet's posterior update is equivalent to predicting a set of pseudo observations $\{\tilde{y}^{(j)}\}_j^{(i)}$ per input $\mathbf{x}^{(i)}$, accordingly $\beta_c^{(i)} = \sum_j \mathbb{1}_{\tilde{y}^{(j)}=c}$ and

$$\mathbb{P}(\mathbf{p}^{(i)}|\{\tilde{y}^{(j)}\}_j^{(i)}) \propto \mathbb{P}(\{\tilde{y}^{(j)}\}_j^{(i)}|\mathbf{p}^{(i)}) \times \mathbb{P}(\mathbf{p}^{(i)}). \tag{3}$$

In practice, we set $\boldsymbol{\beta}^{\text{prior}} = \mathbf{1}$ leading to a flat equiprobable prior when the model brings no additional evidence, i.e. when $\boldsymbol{\beta}^{(i)} = \mathbf{0}$.

The parametrization of $\beta_c^{(i)}$ is crucial and based on two main components. The first component is an encoder neural network, f_θ that maps a data point $\mathbf{x}^{(i)}$ onto a low-dimensional latent vector $\mathbf{z}^{(i)} = f_\theta(\mathbf{x}^{(i)}) \in \mathbb{R}^H$. The second component is to learn a *normalized* probability density $\mathbb{P}(\mathbf{z}|c;\phi)$ per class on this latent space; intuitively acting as class conditionals in the latent space. Given these and the number of ground-truth observations N_c in class c, we define:

$$\beta_c^{(i)} = N_c \cdot \mathbb{P}(\mathbf{z}^{(i)}|c;\phi) = N \cdot \mathbb{P}(\mathbf{z}^{(i)}|c;\phi) \cdot \mathbb{P}(c), \tag{4}$$

which corresponds to the number of (pseudo) observations of class c at $\mathbf{z}^{(i)}$. Note that it is crucial that $\mathbb{P}(\mathbf{z}|c;\phi)$ corresponds to a proper normalized density function since this will ensure the model's epistemic uncertainty to increase outside the known distribution. Indeed, the core idea of our approach is to parameterize these distributions by a flexible, yet tractable family: normalizing flows (e.g. radial flow [34] or IAF [17]). Note that normalizing flows are theoretically capable of modeling any continuous distribution given an expressive and deep enough model [14, 17].

In practice, we observed that various architectures can be used for the encoder. Also, similarly to GAN training [33], we observed that adding a batch normalization after the encoder made the training more stable. It facilitates the match between the latent positions output by the encoder and non-zero density regions learned by the normalizing flows. Remark that we can theoretically use any density estimator on the latent space. We experimentally compared Mixtures of Gaussians (MoG), radial flow [34] and IAF [17]. While all density types performed reasonably well (see Tab. 1 and app.), we observed a better performance of flow-based density estimation in general. We decided to use radial flow for its good trade-off between flexibility, stability, and compactness (only few parameters).

Model discussion. Equation 4 exhibits a set of interesting properties which ensure reasonable uncertainty estimates for ID and OOD samples. To highlight the properties of the Dirichlet distributions learned by PostNet, we assume in this paragraph that we have a fixed encoder f_θ and normalizing flow model parameterized by ϕ. Writing the mean of the Dirichlet distribution parametrized by (4) and using Bayes' theorem gives:

$$\mathbb{E}_{\mathbf{p}\sim\mathrm{Dir}(\boldsymbol{\alpha}^{(i)})}[p_c] = \frac{\beta_c^{\mathrm{prior}} + N \cdot \mathbb{P}(c|\mathbf{z}^{(i)};\phi) \cdot \mathbb{P}(\mathbf{z}^{(i)};\phi)}{\sum_c \beta_c^{\mathrm{prior}} + N \cdot \mathbb{P}(\mathbf{z}^{(i)};\phi)} \tag{5}$$

For very likely in-distribution data (i.e. $\mathbb{P}(\mathbf{z}^{(i)};\phi) \to \infty$), the aleatoric distribution estimate $\bar{\mathbf{p}}^{(i)} = \mathbb{E}_{\mathbf{p}\sim\mathrm{Dir}(\boldsymbol{\alpha}^{(i)})}[p_c]$ converges to the true categorical distribution $\mathbb{P}(c|\mathbf{z}^{(i)};\phi)$. Thus, predictions are more accurate and calibrated for likely samples. Conversely, for out-of-distribution samples (i.e. $\mathbb{P}(\mathbf{z}^{(i)};\phi) \to 0$), the aleatoric distribution estimate $\bar{\mathbf{p}}^{(i)} = \mathbb{E}_{\mathbf{p}\sim\mathrm{Dir}(\boldsymbol{\alpha}^{(i)})}[p_c]$ converges to the flat prior distribution (e.g. $p_c = \frac{1}{C}$ if $\boldsymbol{\beta}^{prior} = \mathbf{1}$). In the same way, we show in the appendix that the covariance of the epistemic distribution converges to $\mathbf{0}$ for very likely in-distribution data, meaning no epistemic uncertainty. Thus, uncertainty for in-distribution data is reduced to the (inherent) aleatoric uncertainty and zero epistemic uncertainty. Similarly to the single categorical distribution case, increasing the training dataset size (i.e. $N \to \infty$) also leads to the mean prediction $\mathbb{E}_{\mathbf{p}\sim\mathrm{Dir}(\boldsymbol{\alpha})}[p_c]$ converging to the class posterior $\mathbb{P}(c|\mathbf{z}^{(i)};\phi)$. On the contrary, no observed data (i.e $N = 0$) again leads to the model reverting to a flat prior.

Posterior Network also handles limited certainty budgets at different levels. At the sample level, the certainty budget $\alpha_0^{(i)} = \sum_c \alpha_c^{(i)}$ is distributed over classes. At the class level, the certainty budget $N_c = \int N_c \, \mathbb{P}(\mathbf{z}|c;\phi) \, d\mathbf{z} = N_c \int \mathbb{P}(\mathbf{z}|c;\phi) \, d\mathbf{z}$ is distributed over samples. At the dataset level, the certainty budget $N = \sum_c \int N_c \, \mathbb{P}(\mathbf{z}|c;\phi) \, d\mathbf{z}$ is distributed over classes and samples. Regions of latent space with many training examples are assigned high density $\mathbb{P}(\mathbf{z}|c;\phi)$, forcing low density elsewhere to fulfill the integration constraint. Consequently, density estimation using normalizing flows enables PostNet to learn out-of-distribution uncertainty by observing only in-distribution data.

Overview. In Figure 2 we provide an overview of Posterior Network. We have three example inputs, $\mathbf{x}^{(1)}$, $\mathbf{x}^{(2)}$, and $\mathbf{x}^{(3)}$, which are mapped onto their respective latent space coordinates $\mathbf{z}^{(i)}$ by the encoding neural network f_θ. The normalizing flow component learns flexible (normalized) density functions $\mathbb{P}(\mathbf{z}|c;\phi)$, for which we evaluate their densities at the positions of the latent vectors $\mathbf{z}^{(i)}$. These densities are used to parameterize a Dirichlet distribution for each data point, as seen on the right hand side. Higher densities correspond to higher confidence in the Dirichlet distributions – we can observe that the out-of-distribution sample $\mathbf{x}^{(3)}$ is mapped to a point with (almost) no density, and hence its predicted Dirichlet distribution

Figure 2: Overview of Posterior Network.

has very high epistemic uncertainty. On the other hand, $\mathbf{x}^{(2)}$ is an ambiguous example that could depict either the digit 0 or 6. This is reflected in its corresponding Dirichlet distribution, which has high aleatoric uncertainty (as the sample is ambiguous), but low epistemic uncertainty (since it is from the distribution of hand-drawn digits). The unambiguous sample $\mathbf{x}^{(1)}$ has low overall uncertainty.
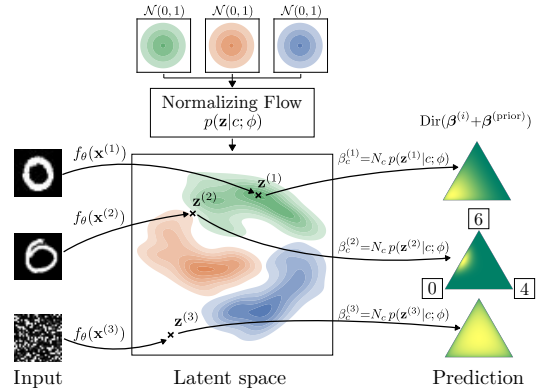
Lastly, since both the encoder network f_θ and the normalizing flow parameterized by ϕ are fully differentiable, we can learn their parameters jointly in an end-to-end fashion. We do this via a novel loss defined in Sec.3 which emerges from Bayesian learning principles [36] and is related to UCE [1].

2.2 Density estimation for OOD detection

Normalized densities, as used by PostNet, are well suited to discriminate between ID data (with high likelihoood) and OOD data (with low likelihood). While it is also possible to learn a normalizing flow model on the input domain directly (e.g., [31, 15, 9]), this is very computationally demanding and might not be necessary for a discriminative model. Futhermore, density estimation is prone to the curse of dimensionality in high dimensional spaces [27, 4]. For example, unsupervised deep

generative models like [16] or [39] have been shown to be unable to distinguish between ID and OOD samples in some situations when working on all features directly [28, 10].

To circumvent these issues, PostNet leverages two techniques. First, it uses the full class label information. Thus, PostNet assigns a density per class and regularizes the epistemic uncertainty with the training class counts N_c. Second, PostNet performs density estimation on a low dimensional latent space describing relevant features for the classification task (Fig. 1; Fig. 2). Hence, PostNet does not suffer from the curse of dimensionality and still enjoys the benefits of a properly normalized density function. Using the inductive bias of a discriminative task and low dimensional latent representations improved OOD detection in [18] as well.

3 Uncertainty-Aware Loss Computation

A crucial design choice for neural network learning is the loss function. PostNet estimates both aleatoric and epistemic uncertainty by learning a distribution $q^{(i)}$ for data point i which is close to the true posterior of the categorical distribution $\mathbf{p}^{(i)}$ given the training data $\mathbf{x}^{(i)}$: $q(\mathbf{p}^{(i)}) \simeq \mathbb{P}(\mathbf{p}^{(i)}|\mathbf{x}^{(i)})$. One way to approximate the posterior is the following Bayesian loss function [2, 36, 41], which has the nice property of being optimal when $q^{(i)}$ is equal to the true posterior distribution:

$$q^* = \underset{q^{(i)} \in \mathcal{P}}{\arg\min} \; \mathbb{E}_{\psi^{(i)} \sim q(\psi^{(i)})}[l(\psi^{(i)}, \mathbf{x}^{(i)})] - H(q^{(i)}), \tag{6}$$

where l is a generic loss over $\psi^{(i)}$ satisfying $0 < \int \exp(-l(\psi, x))d\psi < \infty$, $\mathcal{P}$ is the family of distributions we consider and $H(q^{(i)})$ denotes the entropy of $q^{(i)}$.

Applied in our case, Posterior Network learns a distribution q from the family of the Dirichlet distributions $\mathrm{Dir}(\boldsymbol{\alpha}^{(i)}) = \mathcal{P}$ over the parameters $\mathbf{p}^{(i)} = \psi^{(i)}$. Instantiating the loss l with the cross-entropy loss (CE) we obtain the following optimization objective

$$\min_{\theta,\phi} \mathcal{L} = \min_{\theta,\phi} \frac{1}{N} \sum_i^N \underbrace{\mathbb{E}_{q(\mathbf{p}^{(i)})}[\mathrm{CE}(\mathbf{p}^{(i)}, \mathbf{y}^{(i)})]}_{(1)} - \underbrace{H(q^{(i)})}_{(2)} \tag{7}$$

where $\mathbf{y}^{(i)}$ corresponds to the one-hot encoded ground-truth class of data point i. Optimizing this loss approximates the true posterior distribution for the the categorical distribution $\mathbf{p}^{(i)}$. The first term (1) corresponds the Uncertain Cross Entropy loss (UCE) introduced by [1], which is known to increase confidence for observed data. The second term (2) is an entropy regularizer, which emerges naturally and favors smooth distributions $q^{(i)}$. Here we are optimizing jointly over the neural network parameters θ and ϕ. We also experimented with sequential training i.e. optimizing over the normalizing flow component only with a pre-trained model (see Tab. 2 and app.). Once trained, PostNet can predict uncertainty-aware Dirichlet distributions for unseen data points.

Observe that Eq. (7) is equivalent to the ELBO loss used in variational inference when using a uniform Dirichlet prior (i.e. $(1) = -\mathbb{E}_{q(\mathbf{p}^{(i)})}[\log \mathbb{P}(\mathbf{y}^{(i)}|\mathbf{p}^{(i)})]$ and $(2) = \mathrm{KL}(q^{(i)}||\mathbb{P}(\mathbf{p}^{(i)}))$ where $\mathbb{P}(\mathbf{p}^{(i)}) = \mathrm{Dir}(\mathbf{1})$). The more general Eq. (6), however, is not necessarily equal to an ELBO loss.

Another interesting special case is to consider the family of Dirac distributions as $\mathcal{P}$ instead of the family of Dirichlet distributions. In this case we find back the traditional cross-entropy loss, which performs a simple point estimate for the distribution $\mathbf{p}^{(i)}$. CE is therefore not suited to learn a distribution with non-zero variance, as explained in [1].

Other approaches, such as dropout, approximate the expectation in UCE by sampling from $q^{(i)}$. Our approach has the advantage of using closed-form expressions both for UCE [1] and the entropy term, thus being efficient and exact. The weight of the entropy regularization is a hyperparameter; experiments have shown PostNet to be fairly insensitive to it, so in our experiments we set it to 10^{-5}.

4 Experimental Evaluation

In this section we compare our model to previous methods on a rich set of experiments. The code and further supplementary material is available online (www.daml.in.tum.de/postnet).

Baselines. We have special focus on comparing with other models parametrizing Dirichlet distributions. We use Prior Networks (PN) trained with KL divergence (**KL-PN**) [24] and Reverse KL divergence (**RKL-PN**) [25]. These methods assume the knowledge of in- and out-of-distribution samples. For fair evaluation, the actual OOD test data cannot be used; instead, we used uniform noise on the valid domain as OOD training data. Additionally we trained RKL-PN with FashionMNIST as OOD data for MNIST (**RKL-PN w/ F.**). We also compare to Distribution Distillation (**Distill.**) [26], which learns Dirichlet distributions with maximum likelihood by using soft-labels from an ensemble of networks. As further baselines, we compare to dropout models (**Dropout Net**) [8] and ensemble methods (**Ensemble Net**) [21], which are state of the art in many tasks involving uncertainty estimation [38]. Empirical estimates of the mean and variance of $q^{(i)}$ are computed based on the neuron drop probability p_{drop}, and m individually trained networks for ensemble.

All models share the same core architecture using 3 dense layers for tabular data, and 3 conv. + 3 dense layers for image data. Similarly to [24, 25], we also used the VGG16 architecture [37] on CIFAR10. We performed a grid search on p_{drop}, m, learning rate and hidden dimensions, and report results for the best configurations. Results are obtained from 5 trained models with different initializations. Moreoever, for all experiments, we split the data into train, validation and test set (60%, 20%, 20%) and train/evaluate all models on 5 different splits. Besides the mean we also report the standard error of the mean. Further details are given in the appendix.

Datasets. We evaluate on the following real-world datasets: **Segment** [6], **Sensorless Drive** [6], **MNIST** [22] and **CIFAR10** [19]. The former two datasets (Segment and Sensorless Drive) are tabular datasets with dimensionality 18 and 49 and with 7 and 11 classes, respectively. We rescale all inputs between $[0, 1]$ by using the min and max value of each dimension from the training set. Additionally, we compare all models on 2D synthetic data composed of three Gaussians each. Datasets are presented in more detail in the appendix.

Uncertainty Metrics. We follow the method proposed in [38] and evaluate the coherence of confidence, uncertainty calibration and OOD detection. Note that our goal is not to improve accuracy; still we report the numbers in the experiments.

<u>Confidence calibration:</u> We aim to answer '*Are more confident (i.e. less uncertain) predictions more likely to be correct?*'. We use the area under the precision-recall curve (AUC-PR) to measure confidence calibration. For aleatoric confidence calibration (**Alea. Conf.**) we use $\max_c \bar{\mathbf{p}}_c^{(i)}$ as the scores with labels 1 for correct and 0 for incorrect predictions. For epistemic confidence calibration (**Epist. Conf.**), we distinguish Dirichlet-based models, and dropout and ensemble models. For the former we use $\max_c \boldsymbol{\alpha}_c^{(i)}$ as scores, and for the latter we use the (inverse) empirical variance $\tilde{\mathbf{p}}_c^{(i)}$ of the predicted class, estimated from 10 samples.

<u>Uncertainty calibration:</u> We used Brier score (**Brier**), which is computed as $\frac{1}{N} \sum_i^N \|\bar{\mathbf{p}}^{(i)} - \mathbf{y}^{(i)}\|_2$, where $\mathbf{y}^{(i)}$ is the one-hot encoded ground-truth class of data point i. For Brier score, lower is better.

<u>OOD detection:</u> Our main focus lies on the models' ability to detect out-of-distribution samples. We used AUC-PR to measure performance. For aleatoric OOD detection (**Alea. OOD**), the scores are $\max_c \bar{\mathbf{p}}_c^{(i)}$ with labels 1 for ID data and 0 for OOD data. Fo epistemic OOD detection (**Epist. OOD**), the scores for Dirichlet-based models are given by $\alpha_0^{(i)} = \sum_c \alpha_c^{(i)}$, while we use the (inverse) empirical variance $\tilde{\mathbf{p}}^{(i)}$ for ensemble and dropout models. To provide a comprehensive overview of OOD detection results we use different types of OOD data as described below.

Unseen datasets. We use data from other datasets as OOD data for the image-based models. We use data from FashionMNIST [40] and K-MNIST [5] as OOD data for models trained on MNIST, and data from SVHN [29] as OOD for CIFAR10.

Left-out classes. For the tabular datasets (Segment and Sensorless Drive) there are no other datasets that are from the same domain. To simulate OOD data we remove one or more classes from the training data and instead consider them as OOD data. We removed one class (class sky) from the Segment dataset and two classes from Sensorless Drive (class 10 and 11).

Out-of-domain. In this novel evaluation we consider an extreme case of OOD data for which the data comes from different value ranges (**OODom**). E.g., for images we feed unscaled versions in the range $[0, 255]$ instead of scaled versions in $[0, 1]$. We argue that models should easily be able to detect data that is extremely far from the data distribution. However, as it turns out, this is surprisingly difficult for many baseline models.

	Acc.	Alea. Conf.	Epist. Conf.	Brier	OOD Alea.	OOD Epist.
Drop Out	89.32±0.2	98.21±0.1	95.24±0.2	28.86±0.4	35.41±0.4	40.61±0.7
Ensemble	99.37±0.0	99.99±0.0	*99.98±0.0	2.47±0.1	50.01±0.0	50.62±0.1
Distill.	93.66±1.5	98.29±0.5	98.15±0.5	44.94±1.4	32.1±0.6	31.17±0.2
KL-PN	94.77±0.9	99.52±0.1	99.47±0.1	21.47±1.9	35.48±0.8	33.2±0.6
RKL-PN	99.42±0.0	99.96±0.0	99.89±0.0	9.07±0.1	45.89±1.6	38.14±0.8
PostN Rad.	98.02±0.1	99.89±0.0	99.47±0.0	5.51±0.2	72.89±0.8	*88.73±0.5
PostN IAF	*99.52±0.0	*100.0±0.0	99.92±0.0	*1.43±0.1	*82.96±0.8	88.65±0.4

Table 1: Results on Sensorless Drive dataset. Bold numbers indicate best score among Dirichlet parametrized models and starred numbers indicate best scores among all models.

(a) PostNet: No-Flow   (b) PostNet: No-Bayes-Loss   (c) PostNet: Complete
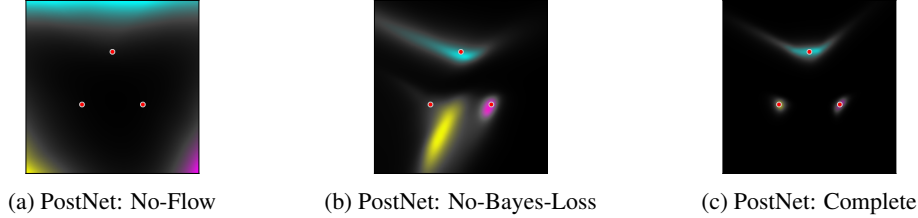
Figure 3: Uncertainty visualization for a 2D 3-Gaussians dataset. Red dots indicate the Gaussians means. Darker regions indicate high epistemic uncertainty for a class prediction. Ablated models fail even a simple dataset while PostNet shows high certainty around gaussians means only.

Dataset shifts. Finally, for CIFAR10, we use 15 different image corruptions at 5 different severity levels [13]. This setting evaluates the models' ability to detect low-quality data (Fig. 4b,c).

Results. Results for the Sensorles Drive dataset are shown in Tab. 1. Tables for other datasets are in the appendix. Even without requiring expensive sampling, PostNet performs on par for accuracy and confidence scores with other models, brings a significant improvement for calibration within the Dirichlet-based models, and outperforms all other models by a large margin (more than $+30\%$ abs. improvement) for OOD detection. Radial flow and IAF variants both achieve strong performance for all datasets (see app.). We use the smaller model (i.e. Radial flow) for comparison in the following. In our experiments, note that using one Radial flow per class represents a small overhead of only 80 parameters per class, which is negligible compared to the encoder architectures (e.g. VGG16 has 138M parameters).

	Acc.	Alea. Conf.	Epist. Conf.	Brier	OOD Alea.	OOD Epist.
PostN: No-Flow	55.38±0.7	85.46±0.3	82.58±0.6	64.4±0.6	29.59±0.1	31.15±0.4
PostN: No-Bayes-Loss	96.6±0.2	99.74±0.0	98.68±0.1	8.85±0.4	62.39±1.5	82.63±1.4
PostN: Seq-No-Bn	15.09±1.0	39.88±7.2	39.86±7.2	89.88±1.3	57.19±2.5	56.74±2.4
PostN: Seq-Bn	98.42±0.1	99.92±0.0	98.76±0.1	5.41±0.1	52.35±0.7	71.75±1.9

Table 2: Ablation study results on Sensorless Drive dataset. Gray cells indicate significant drops in scores compared to complete PostNet Rad. in Tab. 1.

We performed an ablation study on each component of PostNet to evaluate their individual contributions. We were especially interested in comparing stability and uncertainty estimates. Thus, we removed independently the normalizing flow component (No-Flow) and the novel Bayesian loss (No-Bayes-Loss) replaced by the classic cross-entropy loss. Furthermore, we used pre-trained models and subsequently only trained the normalizing flow component, with or without a batch normalization layer (Seq-Bn and Seq-No-Bn). We report results in Tab. 2. No-Flow has a significant drop in OOD detection scores similarly to Prior Networks; not surprising since they mainly differ by their loss. This underlines the importance of using normalized density estimation to differentiate ID and OOD data. The lower performance of No-Bayes-Loss compared to the original model indicates the benefit of using our Bayesian loss. Seq-Bn obtains good performance for some of the metrics, which as a by-product, allows to estimate uncertainty on pre-trained models. Though, we noticed better performance for joint training in general. As shown by Seq-No-Bn scores, the batch normalization

layer brings stability. It intuitively facilitates predicted latent positions to lie on non-zero density regions. Similar conclusions can be drawn on the toy dataset (see Fig. 3) and the Segment dataset (see app.). We further compare various density types and latent dimensions in appendix. We noticed that a too high latent dimension leads to a performance decrease. We also observed that flow-based density estimation generally achieves better scores.

	OOD K. Alea.	OOD K. Epist.	OOD F. Alea.	OOD F. Epist.	OODom K. Alea.	OODom K. Epist.	OODom F. Alea.	OODom F. Epist.
RKL-PN	60.76±2.9	53.76±3.4	78.45±3.1	72.18±3.6	9.35±0.1	8.94±0.0	9.53±0.1	8.96±0.0
RKL-PN w/ F.	81.34±4.5	78.07±4.8	**100.0±0.0**	**100.0±0.0**	9.24±0.1	9.08±0.1	88.96±4.4	87.49±5.0
PostN	**95.75±0.2**	**94.59±0.3**	97.78±0.2	97.24±0.3	**100.0±0.0**	**100.0±0.0**	**100.0±0.0**	**100.0±0.0**

Table 3: Results on MNIST for OOD detection against KMNIST (K.) and FashionMNIST (F.). We trained Rev. KL divergence PriorNets with uniform noise (RKL-PN) and Fashion MNIST (RKL-PN w/ F.) as OOD. PostNet requires no OOD data. Larger numbers are better.

Results of the comparison between RKL-PN, RKL-PN w/ F and PostNet for OOD detection on MNIST are shown in Tab. 3. Not surprisingly, the usage of FashionMNIST as OOD data for training helped RKL-PN to detect other FashionMNIST data. Except for FashionMNIST OOD, PostNet still outperforms RKL-PN w/ F. in OOD detection for other datasets. We noticed that tabular datasets, defined on an unbounded input domain, are more difficult for baselines. One explanation is that due to the $\min/\max$ normalization it can happen that test samples lie outside the interval $[0, 1]$ observed during training. For images, the input domain is compact, which allows to define a valid distribution for OOD data (e.g. uniform) which makes OODom data challenging (see OOD vs OODom in Tab. 3).

	Acc.	Alea. Conf.	Epist. Conf.	Brier	OOD Alea.	OOD Epist.	OODom Alea.	OODom Epist.
Drop Out C.	71.73±0.2	92.18±0.1	84.38±0.3	49.76±0.2	**72.94±0.3**	41.68±0.5	28.3±1.8	47.1±3.3
KL-PN C.	48.84±0.5	78.01±0.6	77.99±0.7	83.11±0.6	59.32±1.1	58.03±0.8	17.79±0.0	20.25±0.2
RKL-PN C.	62.91±0.3	85.62±0.2	81.73±0.2	58.12±0.4	67.07±0.4	56.64±0.8	17.83±0.0	17.76±0.0
PostN C.	**76.46±0.3**	**94.75±0.1**	**94.34±0.1**	**37.39±0.4**	72.83±0.6	**72.82±0.7**	**100.0±0.0**	**100.0±0.0**
Drop Out V.	82.84±0.1	97.15±0.0	96.6±0.0	27.15±0.2	51.39±0.1	53.64±0.1	51.38±0.1	53.66±0.1
KL-PN V.	27.46±1.7	50.61±4.0	52.49±4.2	87.28±1.0	43.96±1.9	43.23±2.3	18.14±0.1	19.12±0.4
RKL-PN V.	64.76±0.3	86.11±0.4	85.59±0.3	54.73±0.4	53.61±1.1	49.37±0.8	29.07±2.1	24.84±1.3
PostN V.	**84.85±0.0**	**97.76±0.0**	**97.25±0.0**	**22.84±0.0**	**80.21±0.2**	**77.71±0.3**	**91.35±0.5**	**99.25±0.1**

Table 4: Results on CIFAR10 with simple convolutional architectures (C.) and VGG16 (V.). Bold numbers indicate best score among one architecture type.

Uncertainty estimation should be good regardless of the model accuracy. It is even more important for less accurate models since they actually *do not know* (i.e. they do more mistakes). Thus, we compared the models that use a single network for training (using a convolutional architecture and VGG16) in Tab. 4. Without the knowledge of true OOD data (SVHN) during training, Prior Networks struggle to achieve good performance. In contrast, PostNet outputs high quality uncertainty estimates regardless of the architecture used for the encoder. We report additional results for PostNet using other encoder architectures (convolutional architecture, AlexNet [20], VGG [37] and ResNet [12]) in Table 5. Deep generative models as Glow [16] using density estimation on input space are unable to distinguish between CIFAR10 and SVHN [28]. In contrast, PostNet clearly distinguishes between in-distribution data (CIFAR10) with low entropy, out-of-distribution (OOD SVHN) with high entropy, and close to the maximum possible entropy for out-of-domain data (OODom SVHN) (see Fig. 4a). Similar conclusions hold for MNIST and FashionMNIST (see app.). Furthermore, results for the image perturbations on CIFAR10 introduced by [13] are presented in Fig. 4. We define the average change in confidence as the ratio between the average confidence $\frac{1}{N} \sum_i^N \alpha_0^{(i)}$ at severity 1 vs other severity levels. As larger shifts correspond to larger differences in the underlying distributions, we expect uncertainty-aware models to become less certain for more severe perturbations. Posterior Network exhibits, as desired, the largest decrease in confidence with stronger corruptions (see Fig. 4b) while maintaining a high accuracy (see Fig. 4c).

	Acc.	Alea. Conf.	Epist. Conf.	Brier	OOD Alea.	OOD Epist.	OODom Alea.	OODom Epist.
PostNet: Conv.	78.58±0.1	95.45±0.0	93.36±0.0	33.84±0.2	72.21±0.1	57.72±0.7	100.0±0.0	100.0±0.0
PostNet: Alexnet	80.81±0.2	96.33±0.1	95.35±0.1	29.99±0.3	73.4±0.7	67.05±0.6	97.64±0.4	99.64±0.1
PostNet: VGG	84.85±0.0	97.76±0.0	97.25±0.0	22.84±0.0	80.21±0.2	77.71±0.3	91.35±0.5	99.25±0.1
PostNet: Resnet	87.86±0.2	98.35±0.0	97.13±0.0	19.33±0.3	79.92±0.4	72.25±0.6	99.94±0.0	99.94±0.0

Table 5: Results of PostNet with different encoder architectures. It shows good uncertainty estimation regardless of the architecture complexity.

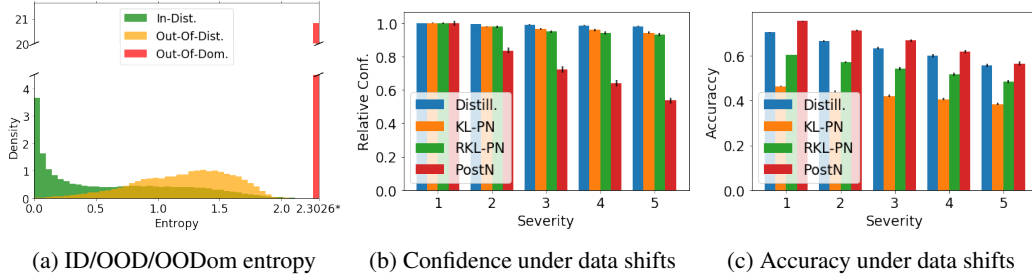(a) ID/OOD/OODom entropy   (b) Confidence under data shifts   (c) Accuracy under data shifts

Figure 4: (a) shows entropy of the aleatoric distributions predicted by PostNet on CIFAR10 (ID) and SVHN (OOD, OODom). The value 2.3026* denotes the highest achievable entropy for 10 classes. PostNet can easily distinguish between the three data types. (b) and (c) present averaged confidence and accuracy under 15 dataset shifts introduced by [13] on CIFAR10 with conv. architecture. On more severe perturbations (i.e. data further away from data distribution), PostNet assigns higher epistemic uncertainty as desired. Baselines keeps same confidence even for less accurate predictions.

5   Conclusion

We propose Posterior Network, a model for uncertainty estimation in classification without requiring out-of-distribution samples for training or costly sampling for uncertainty estimation. PostNet is composed of three main components: an encoder which outputs a position in a latent space, a normalizing flow which performs a density estimation in this latent space, and a Bayesian loss for uncertainty-aware training. In our extensive experimental evaluation, PostNet achieves state-of-the-art performance with a strong improvement for detection of in- and out-of-distribution samples.

Broader Impact

Traditional classification models without uncertainty estimate can be dangerous when used without domain expertise. They might have indeed unexpected behavior in new anomalous situations/input and are unaware of the underlying risk of their predictions. Uncertainty aware models like PostNet try to mitigate the risk of such autonomous predictions by attaching a confidence score to their predictions. On one hand, uncertainty aware predictions could be particularly beneficial in domains with potential critical consequences and prone to automation (e.g. finance, autonomous driving or medicine). When applied, these models are able to refrain from predicting if the data is out of their domain of expertise. Posterior Network makes a significant step further in this direction by even not requiring to observe similar anomalous situations during training. Anomalous data are typically not known in advance since they are rare by definition. Thus Posterior Network significantly increases the applicability of uncertainty estimation across application domains. On the other hand, high-quality of uncertainty estimation might also give a false sense of security. A potential risk is that an excessive trust in the model behavior leads to a lack of human supervision. For example in medicine, predictions wrongly deemed safe could have dramatic repercussions without human control.

Acknowledgments and Disclosure of Funding

References

[1] Marin Biloš, Bertrand Charpentier, and Stephan Günnemann. Uncertainty on asynchronous time event prediction. In *Advances in Neural Information Processing Systems*, pages 12831–12840. 2019.

[2] P. G. Bissiri, C. C. Holmes, and S. G. Walker. A general framework for updating belief distributions. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 2016.

[3] Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight uncertainty in neural networks. In *International Conference on International Conference on Machine Learning*, page 1613–1622, 2015.

[4] Hyunsun Choi and Eric Jang. Generative ensembles for robust anomaly detection, 2019.

[5] Tarin Clanuwat, Mikel Bober-Irizar, Asanobu Kitamoto, Alex Lamb, Kazuaki Yamamoto, and David Ha. Deep learning for classical japanese literature, 2018.

[6] Dheeru Dua and Casey Graff. UCI machine learning repository, 2017.

[7] Dhivya Eswaran, Stephan Günnemann, and Christos Faloutsos. The power of certainty: A dirichlet-multinomial model for belief propagation. In *International Conference on Data Mining*, pages 144–152, 2017.

[8] Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *International Conference on Machine Learning*, pages 1050–1059, 2016.

[9] Will Grathwohl, Ricky T. Q. Chen, Jesse Bettencourt, and David Duvenaud. Scalable reversible generative models with free-form continuous dynamics. In *International Conference on Learning Representations*, 2019.

[10] Will Grathwohl, Kuan-Chieh Wang, Joern-Henrik Jacobsen, David Duvenaud, Mohammad Norouzi, and Kevin Swersky. Your classifier is secretly an energy based model and you should treat it like one. In *International Conference on Learning Representations*, 2020.

[11] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks. In *International Conference on Machine Learning*, pages 1321–1330, 2017.

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385*, 2015.

[13] Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. In *International Conference on Learning Representations*, 2019.

[14] Chin-Wei Huang, David Krueger, Alexandre Lacoste, and Aaron Courville. Neural autoregressive flows. In *International Conference on Machine Learning*, pages 2078–2087, 2018.

[15] Durk P Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. In *Advances in Neural Information Processing Systems*, pages 10215–10224. 2018.

[16] Durk P Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. In *Advances in Neural Information Processing Systems*, pages 10215–10224. 2018.

[17] Durk P Kingma, Tim Salimans, Rafal Jozefowicz, Xi Chen, Ilya Sutskever, and Max Welling. Improved variational inference with inverse autoregressive flow. In *Advances in Neural Information Processing Systems*, pages 4743–4751. 2016.

[18] Polina Kirichenko, Pavel Izmailov, and Andrew Gordon Wilson. Why Normalizing Flows Fail to Detect Out-of-Distribution Data. In *Advances in Neural Information Processing Systems*. 2020.

[19] Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, 2009.

[20] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, pages 1097–1105. 2012.

[21] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in Neural Information Processing Systems 30*, pages 6402–6413. 2017.

[22] Yann LeCun and Corinna Cortes. MNIST handwritten digit database. 2010.

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

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

[25] Andrey Malinin and Mark Gales. Reverse kl-divergence training of prior networks: Improved uncertainty and adversarial robustness. In *Advances in Neural Information Processing Systems*, pages 14520–14531. 2019.

[26] Andrey Malinin, Bruno Mlodozeniec, and Mark Gales. Ensemble distribution distillation. In *International Conference on Learning Representations*, 2020.

[27] Eric Nalisnick, Akihiro Matsukawa, Yee Whye Teh, and Balaji Lakshminarayanan. Detecting out-of-distribution inputs to deep generative models using typicality, 2020.

[28] Eric T. Nalisnick, Akihiro Matsukawa, Yee Whye Teh, Dilan Görür, and Balaji Lakshminarayanan. Do deep generative models know what they don't know? In *International Conference on Learning Representations*, 2019.

[29] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y. Ng. Reading digits in natural images with unsupervised feature learning. In *NIPS Workshop on Deep Learning and Unsupervised Feature Learning*, 2011.

[30] Kazuki Osawa, Siddharth Swaroop, Mohammad Emtiyaz E Khan, Anirudh Jain, Runa Eschenhagen, Richard E Turner, and Rio Yokota. Practical deep learning with bayesian principles. In *Advances in Neural Information Processing Systems*, pages 4289–4301. 2019.

[31] George Papamakarios, Theo Pavlakou, and Iain Murray. Masked autoregressive flow for density estimation. In *Advances in Neural Information Processing Systems*, pages 2338–2347. 2017.

[32] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, pages 8024–8035. 2019.

[33] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. In *International Conference on Learning Representations*, 2016.

[34] Danilo Rezende and Shakir Mohamed. Variational inference with normalizing flows. In *International Conference on Machine Learning*, pages 1530–1538, 2015.

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

[36] John Shawe-Taylor and Robert C. Williamson. A pac analysis of a bayesian estimator. In *Conference on Computational Learning Theory*, page 2–9, New York, NY, USA, 1997.

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

[38] Jasper Snoek, Yaniv Ovadia, Emily Fertig, Balaji Lakshminarayanan, Sebastian Nowozin, D. Sculley, Joshua Dillon, Jie Ren, and Zachary Nado. Can you trust your modeĺs uncertainty? evaluating predictive uncertainty under dataset shift. In *Advances in Neural Information Processing Systems*, pages 13969–13980. 2019.

[39] Aaron van den Oord, Nal Kalchbrenner, Lasse Espeholt, koray kavukcuoglu, Oriol Vinyals, and Alex Graves. Conditional image generation with pixelcnn decoders. In *Advances in Neural Information Processing Systems*, pages 4790–4798. 2016.

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

[41] Arnold Zellner. Optimal information processing and bayes's theorem. *The American Statistician*, page 278, 11 1988.

A Dirichlet Distribution Computations

A.1 Dirichlet distribution

The Dirichlet distribution with concentration parameters $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_C)$, where $\alpha_c > 0$, has the probability density function:

$$f(\boldsymbol{x}; \boldsymbol{\alpha}) = \frac{\prod_{c=1}^{C} \Gamma(\alpha_c)}{\Gamma\left(\sum_{c=1}^{C} \alpha_c\right)} \prod_{c=1}^{C} x_i^{\alpha_c - 1} \tag{8}$$

where Γ is a gamma function:

$$\Gamma(\alpha) = \int_0^\infty \alpha^{z-1} e^{-\alpha} dz$$

A.2 Closed-form formula for Bayesian loss.

The novel Bayesian loss described in formula 7 can be computed in closed form. For the sample $\mathbf{x}^{(i)}$, it is given by:

$$\mathcal{L}^{(i)} = \underbrace{\mathbb{E}_{q(\mathbf{p}^{(i)})}[\text{CE}(\mathbf{p}^{(i)}, \mathbf{y}^{(i)})]}_{(1)} - \underbrace{H(q^{(i)})}_{(2)} \tag{9}$$

where the distribution q belongs to the family of the Dirichlet distributions $\text{Dir}(\boldsymbol{\alpha}^{(i)})$. The term (1) is the UCE loss [1]. Given that the observed class one-hot encoded by $\mathbf{y}^{(i)}$ is denoted by $c*$, the term (1) is equal to:

$$\mathbb{E}_{q(\mathbf{p}^{(i)})}[\text{CE}(\mathbf{p}^{(i)}, \mathbf{y}^{(i)})] = \Psi(\alpha_{c*}^{(i)}) - \Psi(\alpha_0^{(i)}) \tag{10}$$

where Ψ denotes the digamma function. The term (2) is the entropy of a Dirichlet distribution and is given by:

$$H(q^{(i)}) = \log B(\boldsymbol{\alpha}^{(i)}) + (\alpha_0^{(i)} - C)\Psi(\alpha_0^{(i)}) - \sum_c (\alpha_c^{(i)} - 1)\Psi(\alpha_c^{(i)}) \tag{11}$$

where B denotes the beta function.

A.3 Epistemic covariance for in-distribution samples in PostNet.

The epistemic distribution in PostNet is a Dirichlet distribution $\text{Dir}(\boldsymbol{\alpha}^{(i)})$ with the following concentration parameters $\alpha_c^{(i)} = \beta_c^{\text{prior}} + N \cdot \mathbb{P}(c|\mathbf{z}^{(i)}; \phi) \cdot \mathbb{P}(\mathbf{z}^{(i)}; \phi)$ and $\alpha_0^{(i)} = \sum_c \beta_c^{\text{prior}} + N \cdot \mathbb{P}(\mathbf{z}^{(i)}; \phi)$. We can write the variance:

$$\text{Var}_{\mathbf{p} \sim \text{Dir}(\boldsymbol{\alpha}^{(i)})}(p_c) = \frac{\alpha_c(\alpha_0 - \alpha_c)}{\alpha_0^2(\alpha_0 + 1)}; \ \text{Cov}_{\mathbf{p} \sim \text{Dir}(\boldsymbol{\alpha}^{(i)})}(p_c, p_{c'}) = \frac{-\alpha_c \alpha_{c'}}{\alpha_0^2(\alpha_0 + 1)} \tag{12}$$

For in-distribution data (i.e. $\mathbb{P}(\mathbf{z}^{(i)}; \phi) \to \infty$), we have $\text{Var}_{\mathbf{p} \sim \text{Dir}(\boldsymbol{\alpha}^{(i)})}(p_c) = \mathcal{O}\left(\frac{\mathbb{P}(c|\mathbf{z}^{(i)}; \phi)(1 - \mathbb{P}(c|\mathbf{z}^{(i)}; \phi))}{\mathbb{P}(\mathbf{z}^{(i)}; \phi)N}\right)$ and $\text{Cov}_{\mathbf{p} \sim \text{Dir}(\boldsymbol{\alpha}^{(i)})}(p_c, p_{c'}) = \mathcal{O}\left(\frac{-\mathbb{P}(c|\mathbf{z}^{(i)}; \phi)\mathbb{P}(c'|\mathbf{z}^{(i)}; \phi)}{\mathbb{P}(\mathbf{z}^{(i)}; \phi)N}\right)$. Hence both terms converge to 0 when $\mathbb{P}(\mathbf{z}^{(i)}; \phi) \to \infty$.

B Model details

For vector datasets, all models share an architecture of 3 linear layers with Relu activation. A grid search in $[32, 64, 128]$ led to no significant changes in the performances. Therefore, we decided to use 64 hidden units for each layer. For image datasets, we used LeakyRelu activation and add on the top 3 convolutional layers with kernel size of 5, followed by a Max-pooling of size 2. Alternatively, we used the VGG16 architecture with batch normalization [37] adapted from PyTorch implementation [32]. All models are trained after a grid search for learning rate in $[1e^{-3}, 1e^{-5}]$. All models were optimized with Adam optimizer without further learning rate scheduling. We performed early stopping by

checking loss improvement every 2 epochs and a patience equal to 10. We trained all models on GPUs (1TB SSD).

For the dropout models, we used $p_{\text{drop}} = .25$ after a grid search in $[.25, .5, .75]$ and sampled 10 times for uncertainty estimation. As an indicator, the original paper [8], uses a dropout probability of .5 for MNIST. It also states that 10 samples already lead to reasonable uncertainty estimates. For the ensemble models, we used $m = 10$ networks after a grid search in $[2, 5, 10, 20]$. A greater number of networks was also found to give no great improvements in the original paper [21]. To be fair with these models, we distilled the knowledge of 10 neural networks for Distribution Distillation. We also trained Prior Networks where target parameters $\beta_{\text{in}} = 1e^2$ as suggested in original papers [24, 25].

For PostNet, we used a 1D batch normalization after the encoder. Experiments on latent dimensions and density types are presented in following sections. If not explicitley mentioned otherwise, we used by default radial flow with a flow length of 6 and a latent dimension of 6. This leads to only 80 parameters. Comparison with IAF are done with two layers of size 256. In general, we found out that a latent dimension smaller or equal to the number of classes is sufficient. It enables classes to be orthogonal in the latent space if necessary.

All metrics have been scaled by 100. We obtain numbers in $[0, 100]$ for all scores instead of $[0, 1]$.

C Datasets details

For all datasets, we use 5 different random splits to train all models. We split the data in training, validation and test sets $(60\%, 20\%, 20\%)$. In particular, we did not restrict to classic MNIST and CIFAR10 splits in order do prevent overfitting to a specific split.

We use one toy dataset composed of three 2D isotropic Gaussians corresponding to three classes. The Gaussians means are $[0, 2.]$, $[-1.73205081, -1.]$ and $[1.73205081, -1.]$. The variance of the Gaussians is 0.2. A visualization of the true distributions for the three Gaussians is given in Figure 5. The final dataset is composed in total of 1500 samples.

We use the segment vector dataset [6], where the goal is to classify areas of images into 7 classes (window, foliage, grass, brickface, path, cement, sky). We remove the class 'sky' from training and instead use it as the OOD dataset for OOD detection experiments. Each input is composed of 18 attributes describing the image area. The dataset contains $2,310$ samples in total.

We further use the Sensorless Drive vector dataset [6], where the goal is to classify extracted motor current measurements into 11 different classes. We remove classes 10 and 11 from training and use them as the OOD dataset for OOD detection experiments. Each input is composed of 49 attributes describing motor behaviour. The dataset contains $58,509$ samples in total.

Additionally, we use the MNIST image dataset [22] where the goal is to classify pictures of hand-drawn digits into 10 classes (from digit 0 to digit 9). Each input is composed of a $1 \times 28 \times 28$ tensor. The dataset contains $70,000$ samples. For OOD detection experiments, we use KMNIST [5] and FashionMNIST [40] containing images of Japanese characters and images of clothes, respectively.

Finally, we use the CIFAR10 image dataset [19] where the goal is to classify a picture of objects into 10 classes (airplane, automobile, bird, cat, deer, dog, frog, horse, ship, truck). Each input is a $3 \times 32 \times 32$ tensor. The dataset contains $60,000$ samples. For OOD detection experiments, we use street view house numbers (SVHN) [29] containing images of numbers. For the dataset shift experiments, we use the classic split of CIFAR10 to avoid data leakage with the corrupted images from the test set that is provided online.

D Additional Experimental Results

In this section, we present additional results for uncertainty estimation on other datasets. Tables 6, 8, 9, 10, 12 show the performance of all models on the Segment, Sensorless Drive, MNIST, and CIFAR10 datasets. In the same way as for the other datasets, PostNet is competitive for all metrics and show a significant improvement on calibration among Dirichlet parametrized models and on OOD detection tasks among all models. We evaluated the performances of all models on MNIST using different uncertainty measures and observed very correlated results (see Table 11). We compared different encoder architectures on CIFAR10 (see Fig. 5). Without further parameter tuning,

PostNet adapted well to the convolutional architecture, AlexNet [20], VGG [37] and ResNet [12]. For easier comparison, we also trained models on the classic CIFAR10 split (79%, 5%, 16%) with VGG architecture. We noticed that a larger training set leads to better accuracy for all models.

We also show results of experiments with different latent dimensions (see Fig. 7; 8; 13; 14) and density types (MoG, radial, IAF) (see Tab. 6; 8; 9; 10; 12) for all datasets. We remarked that PostNet works with various type of densities even if using mixture of Gaussians presented more instability in practice. We observed no clear winner between Radial flow and IAF. We observed a bit lower performances for MoG which could be explained by its lack of expressiveness. Furthermore, we observed that a too high latent dimension would affect the performance.

Beside tables and figures with detailed metrics, we report additional visualizations. We present the uncertainty visualization on the input space for a 2D toy dataset (see Fig. 6). We show this visualization for all models parametrizing Dirichlet distributions. PostNet is the only model which is not overconfident for OOD data. In particular, it demonstrates the best fit of the true in-distribution data shown in figure 5. Other models show overconfident prediction for OOD regions and fail even on this simple dataset.

Furthermore, we plotted histograms of entropy of ID, OOD, OODom data for MNIST and CIFAR10 (see Fig 10 and 4). For both datasets, PostNet can easily distinguish between the three data types.

Finally, We also included the evolution of the uncertainty while interpolating linearly between images of MNIST (see Fig. 11 and 12). It corresponds to a smooth walk in latent space. As shown in Figure 11, PostNet predicts correctly on clean images and outputs more balanced class predictions for mixed images. Additionally the Figure 12 shows the evolution of the concentration parameters and consequently the epistemic uncertainty. We observe that the epistemic uncertainty (i.e. low α_c) is higher on mixed images which do not correspond to proper digits.
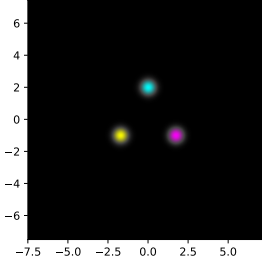

Figure 5: Three Gaussians toy dataset.

	Acc.	Alea. Conf.	Epist. Conf.	Brier	OOD Alea.	OOD Epist.
Drop Out	95.25±0.1	99.75±0.0	99.43±0.0	11.89±0.2	41.48±0.5	43.11±0.6
Ensemble	*97.27±0.1	*99.88±0.0	*99.85±0.0	*7.64±0.2	54.76±1.6	58.13±1.7
Distill.	96.21±0.1	99.82±0.0	**99.8±0.0**	57.77±0.6	37.12±0.5	35.83±0.4
KL-PN	95.61±0.1	99.79±0.0	99.76±0.0	16.84±0.3	65.62±2.4	57.07±3.7
RKL-PN	96.36±0.2	99.71±0.0	99.58±0.0	11.97±0.1	75.46±2.4	51.02±0.6
PostN Rad. (2)	95.76±0.1	99.23±0.1	98.82±0.1	13.33±1.3	92.75±1.3	90.41±1.5
PostN Rad. (6)	96.52±0.2	99.82±0.0	99.43±0.0	8.69±0.3	98.27±0.2	98.09±0.3
PostN Rad. (10)	94.9±0.2	99.51±0.0	98.57±0.1	12.22±0.7	95.53±0.8	97.51±0.7
PostN IAF (2)	93.94±0.3	99.02±0.1	98.3±0.2	15.33±0.7	**98.3±0.3	**99.33±0.1
PostN IAF (6)	95.71±0.2	99.63±0.0	99.11±0.1	10.16±0.3	96.92±0.9	98.17±0.6
PostN IAF (10)	**96.92±0.1**	**99.83±0.0**	99.49±0.0	**8.45±0.4**	95.75±1.1	96.74±0.9
PostN MoG (2)	63.43±5.3	79.61±6.2	79.05±6.1	54.14±5.4	90.87±1.4	91.62±1.4
PostN MoG (6)	89.75±2.5	95.28±1.6	93.15±2.1	24.42±4.3	96.04±1.3	97.71±0.8
PostN MoG (10)	94.44±0.5	99.64±0.1	99.08±0.2	14.79±1.3	91.14±1.5	90.82±1.3

Table 6: Results on Segment dataset with all models. It shows results with different density types. Number into parentheses indicates flow size (for radial flow and IAF) or number of components (for MoG). Bold numbers indicate best score among Dirichlet parametrized models and starred numbers indicate best scores among all models.
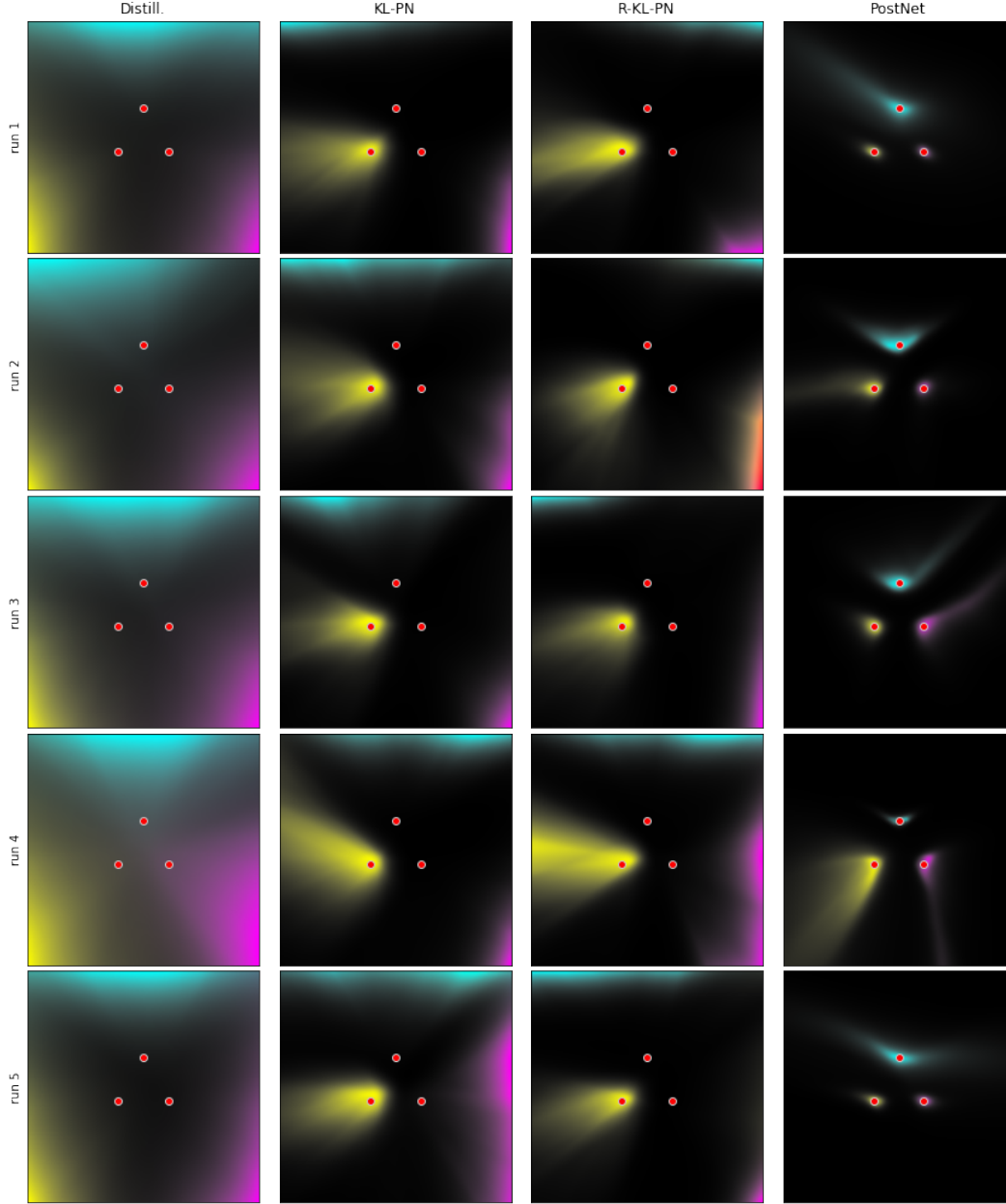
Figure 6: Visualization of the concentration parameters predicted by Distribution Distillation, Prior Networks trained with KL and reverse KL divergence and Posterior Network on a 3-Gaussians toy dataset over 5 runs. Red dots indicate the mean of the 3 Gaussians. Colours indicate class labels predicted by the models, dark regions correspond to high epistemic uncertainty. PostNet consistently predicts low uncertainty around the training data and high uncertainty for OOD data.

	Acc.	Alea. Conf.	Epist. Conf.	Brier	OOD Alea.	OOD Epist.
PostN: No-Flow	93.13±0.3	99.48±0.1	98.41±0.3	12.94±0.3	47.3±2.9	35.49±0.3
PostN: No-Bayes-Loss	93.94±0.8	98.53±0.3	96.08±1.1	16.15±1.9	94.71±1.0	95.92±0.8
PostN: Seq-No-Bn	18.94±1.1	20.42±1.7	20.42±1.7	91.29±0.0	58.91±0.8	58.43±0.8
PostN: Seq-Bn	93.89±0.1	99.38±0.1	98.93±0.0	14.64±0.3	98.02±0.4	99.93±0.0

Table 7: Ablation study results on Segment dataset. Gray cells indicate significant drops in scores compare to the complete PostNet Rad. (6) model in Table 6.
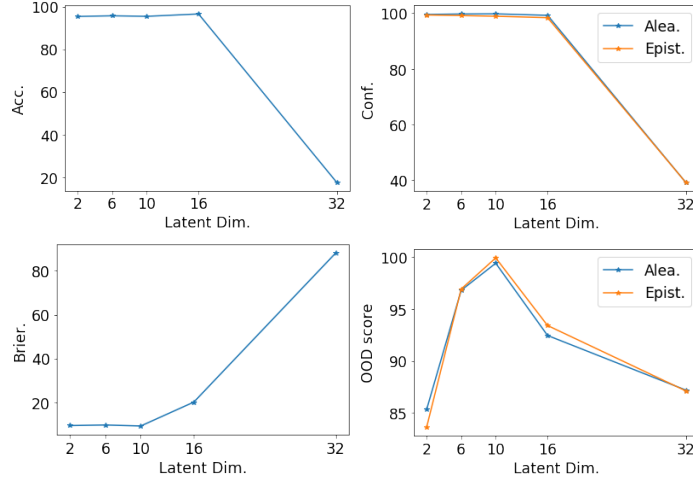
Figure 7: Accuracy and uncertainty scores of PostNet with latent dimension in $[2, 6, 10, 32]$ on the Segment dataset. We observed that the performances remains high for small dimensions (i.e. 2, 6, 10) and drop for a too high dimension (i.e. 32).

	Acc.	Alea. Conf.	Epist. Conf.	Brier	OOD Alea.	OOD Epist.
Drop Out	89.32±0.2	98.21±0.1	95.24±0.2	28.86±0.4	35.41±0.4	40.61±0.7
Ensemble	99.37±0.0	99.99±0.0	*99.98±0.0	2.47±0.1	50.01±0.0	50.62±0.1
Distill.	93.66±1.5	98.29±0.5	98.15±0.5	44.94±1.4	32.1±0.6	31.17±0.2
KL-PN	94.77±0.9	99.52±0.1	99.47±0.1	21.47±1.9	35.48±0.8	33.2±0.6
RKL-PN	99.42±0.0	99.96±0.0	99.89±0.0	9.07±0.1	45.89±1.6	38.14±0.8
PostN Rad. (2)	96.07±0.0	99.28±0.0	98.88±0.0	19.94±0.0	*98.22±0.0	*98.03±0.0
PostN Rad. (6)	98.02±0.1	99.89±0.0	99.47±0.0	5.51±0.2	72.89±0.8	88.73±0.5
PostN Rad. (10)	97.3±0.0	99.82±0.0	99.31±0.0	7.93±0.0	66.65±0.0	87.91±0.0
PostN IAF (2)	99.19±0.0	99.98±0.0	99.78±0.0	2.45±0.0	78.13±0.0	85.9±0.0
PostN IAF (6)	99.11±0.1	99.98±0.0	99.72±0.0	2.71±0.1	78.48±0.7	86.47±0.5
PostN IAF (10)	*99.52±0.0	*100.0±0.0	99.92±0.0	*1.43±0.1	82.96±0.8	88.65±0.4
PostN MoG (2)	59.63±4.8	72.2±4.7	70.38±4.8	68.41±4.6	67.2±3.1	72.3±2.9
PostN MoG (6)	96.83±0.2	99.72±0.0	99.16±0.1	13.24±1.0	59.82±2.3	60.61±2.6
PostN MoG (10)	96.65±0.2	99.64±0.0	99.12±0.1	13.12±1.1	61.54±1.8	65.35±2.0

Table 8: Results on Sensorless Drive dataset with all models. It shows results with different density types. Number into parentheses indicates flow size (for radial flow and IAF) or number of components (for MoG).
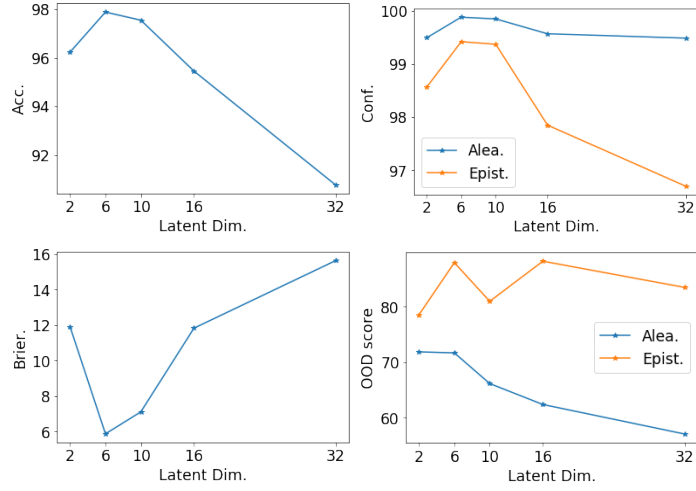
Figure 8: Accuracy and uncertainty scores of PostNet with latent dimension in $[2, 6, 10, 32]$ on the Sensorless Drive dataset. OOD scores are computed against the left out sky class. We observed that the performances remains high for medium dimensions (i.e. 6, 10) and drop for a too high dimension (i.e. 32).

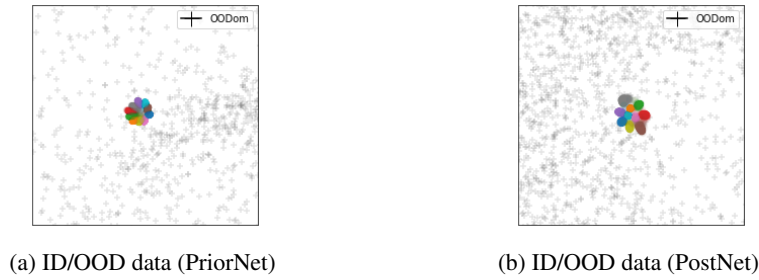(a) ID/OOD data (PriorNet)

(b) ID/OOD data (PostNet)

Figure 9: This figure should be seen in perspective with Fig. 1. We plot FashionMNIST OODom data with black crosses to show where these data would land. OODom data were not used for training the models, A comparison of Fig. 9(a) with Fig. 1(b) show that Prior Network assigns high certainty to OODom data. In contrast, a comparison of Fig. 9(b) and Fig. 1(c) shows that Posterior Network assigns low uncertainty to OODom data as desired.
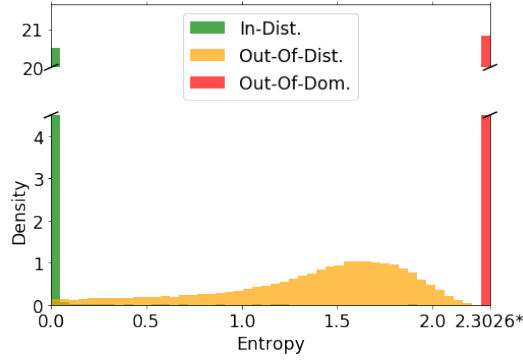
(a) MNIST

Figure 10: Histograms of the entropy of the predicted categorical distributions for in-distribution (green), out-of-distribution (yellow) and out-of-domain (red) data. The value 2.3026^* denotes the maximal entropy achievable for a categorical distribution with 10 classes. We use MNIST, FashionMNIST and the unscaled version of FashionMNIST as in-distribution, out-of-distribution and out-of-domain data. PostNet clearly distinguishes between the three types of data with low entropy for in-distribution data and high entropy for out-of-distribution, and close to the maximum possible entropy for out-of-domain data.
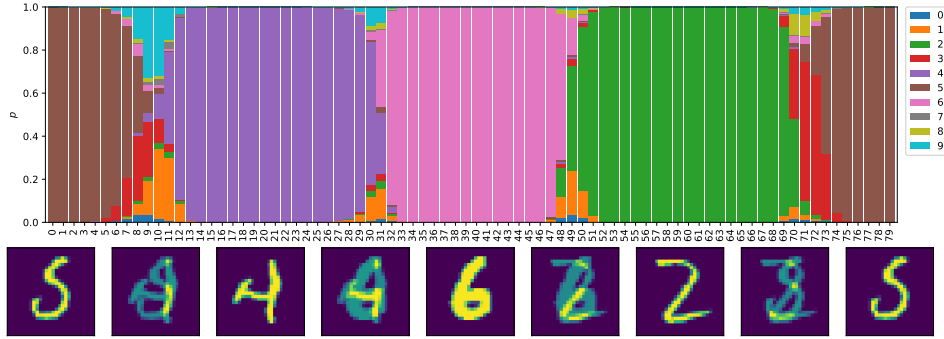

Figure 11: Evolution of the probability predictions when interpolating linearly between four MNIST images. The interpolation goes from the clean digits 5, 4, 6 and 2 in a cyclic way with 20 interpolated images between each pair. As desired, We can observe correct predictions around clean images with higher (aleatoric) uncertainty for mixed images, and smooth transitions in between.
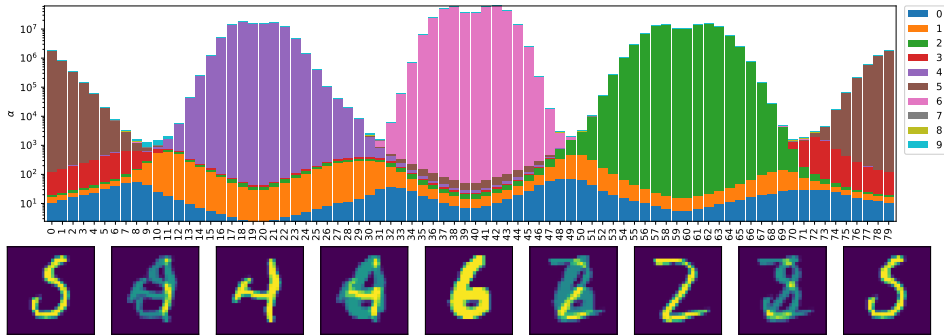

Figure 12: Evolution of the concentration parameters predictions when interpolating linearly between four MNIST images. The interpolation goes from the clean digits 5, 4, 6 and 2 in a cyclic way with 20 interpolated images between each pair. As desired, we can observe correct and confident predictions around clean images with higher (epistemic) uncertainty for mixed images.

18

	Acc.	Alea. Conf.	Epist. Conf.	Brier
Drop Out	99.26±0.0	99.98±0.0	99.97±0.0	1.78±0.0
Ensemble	*99.35±0.0	*99.99±0.0	*99.98±0.0	1.67±0.0
Distill.	**99.34±0.0**	**99.98±0.0**	*99.98±0.0	72.55±0.2
KL-PN	99.01±0.0	99.92±0.0	99.95±0.0	10.82±0.0
RKL-PN	99.21±0.0	99.67±0.0	99.57±0.0	9.76±0.0
RKL-PN w/ F.	99.2±0.0	99.75±0.0	99.68±0.0	9.9±0.0
PostN Rad. (2)	**99.34±0.0**	**99.98±0.0**	99.97±0.0	*1.25±0.0
PostN Rad. (6)	99.28±0.0	99.97±0.0	99.96±0.0	1.36±0.0
PostN Rad. (10)	99.22±0.0	99.97±0.0	99.97±0.0	1.41±0.0
PostN IAF (2)	99.06±0.0	99.96±0.0	99.94±0.0	1.48±0.0
PostN IAF (6)	99.08±0.0	99.96±0.0	99.94±0.0	1.45±0.1
PostN IAF (10)	98.97±0.0	99.96±0.0	99.94±0.0	1.61±0.0
PostN MoG (2)	76.41±2.3	99.93±0.0	99.92±0.0	23.23±2.2
PostN MoG (6)	99.21±0.0	99.94±0.0	99.92±0.0	1.61±0.0
PostN MoG (10)	99.22±0.0	99.94±0.0	99.92±0.0	1.53±0.0

Table 9: Accuracy, confidence and calibration results on MNIST dataset with all models. It shows results with different density types. Number into parentheses indicates flow size (for radial flow and IAF) or number of components (for MoG). Bold numbers indicate best score among Dirichlet parametrized models and starred numbers indicate best scores among all models.

	OOD K. Alea.	OOD K. Epist.	OOD F. Alea.	OOD F. Epist.	OODom K. Alea.	OODom K. Epist.	OODom F. Alea.	OODom F. Epist.
Drop Out	94.0±0.1	93.01±0.2	96.56±0.2	95.0±0.2	31.59±0.5	31.97±0.5	27.2±1.1	27.52±1.1
Ensemble	*97.12±0.0	*96.5±0.0	98.15±0.1	96.76±0.0	41.7±0.3	42.25±0.3	37.22±1.0	37.73±1.0
Distill.	**96.64±0.1**	85.17±1.0	98.83±0.0	94.09±0.4	11.49±0.3	10.66±0.2	13.82±0.5	12.03±0.3
KL-PN	92.97±1.2	93.39±1.0	98.44±0.1	98.16±0.0	9.54±0.1	9.78±0.1	9.57±0.1	10.06±0.1
RKL-PN	60.76±2.9	53.76±3.4	78.45±3.1	72.18±3.6	9.35±0.1	8.94±0.0	9.53±0.1	8.96±0.0
RKL-PN w/ F.	81.34±4.5	78.07±4.8	*100.0±0.0	*100.0±0.0	9.24±0.1	9.08±0.1	88.96±4.4	87.49±5.0
PostN Rad. (2)	95.49±0.3	93.12±0.7	96.2±0.3	94.6±0.4	*100.0±0.0	*100.0±0.0	*100.0±0.0	*100.0±0.0
PostN Rad. (6)	95.75±0.2	**94.59±0.3**	97.78±0.2	97.24±0.3	*100.0±0.0	*100.0±0.0	*100.0±0.0	*100.0±0.0
PostN Rad. (10)	95.46±0.4	94.19±0.4	97.33±0.2	96.75±0.3	*100.0±0.0	*100.0±0.0	*100.0±0.0	*100.0±0.0
PostN IAF (2)	92.24±0.3	91.75±0.3	96.58±0.2	96.6±0.2	*100.0±0.0	*100.0±0.0	*100.0±0.0	*100.0±0.0
PostN IAF (6)	90.74±0.6	90.63±0.6	93.66±0.5	93.17±0.6	*100.0±0.0	*100.0±0.0	*100.0±0.0	*100.0±0.0
PostN IAF (10)	87.08±0.2	86.52±0.3	92.34±0.6	91.27±0.9	*100.0±0.0	*100.0±0.0	*100.0±0.0	*100.0±0.0
PostN MoG (2)	74.27±2.0	73.34±1.9	76.99±2.0	76.74±1.9	*100.0±0.0	*100.0±0.0	99.99±0.0	99.99±0.0
PostN MoG (6)	84.67±1.5	81.46±1.9	88.98±1.7	87.07±2.1	*100.0±0.0	*100.0±0.0	*100.0±0.0	*100.0±0.0
PostN MoG (10)	85.14±1.3	81.12±1.5	94.43±0.8	93.8±1.0	*100.0±0.0	*100.0±0.0	*100.0±0.0	*100.0±0.0

Table 10: OOD results on MNIST dataset with all models. It shows results with different density types. Number into parentheses indicates flow size (for radial flow and IAF) or number of components (for MoG).

	OOD K. α_0/var.	OOD K. MI.	OOD F. α_0/var.	OOD F. MI.	OODom K. α_0/var.	OODom K. MI.	OODom F. α_0/var.	OODom F. MI
Ensemble	*97.19±0.0	*97.44±0.0	97.53±0.1	97.69±0.1	42.36±0.3	42.38±0.3	37.85±1.1	37.86±1.1
RKL-PN	54.11±3.4	54.9±3.3	72.54±3.6	73.33±3.5	8.94±0.0	8.94±0.0	8.96±0.0	8.96±0.0
RKL-PN w/ F.	78.4±4.8	78.73±4.8	*100.0±0.0	*100.0±0.0	9.08±0.1	9.08±0.1	87.49±5.0	87.49±5.0
PostN	**96.04±0.2**	**96.05±0.2**	98.17±0.2	98.17±0.2	*100.0±0.0	*100.0±0.0	*100.0±0.0	*100.0±0.0

Table 11: OOD detection on MNIST with other uncertainty measures. Mutual Information [24] and α_0 (Dirichlet) / variance (Ensemble) results are highly correlated.
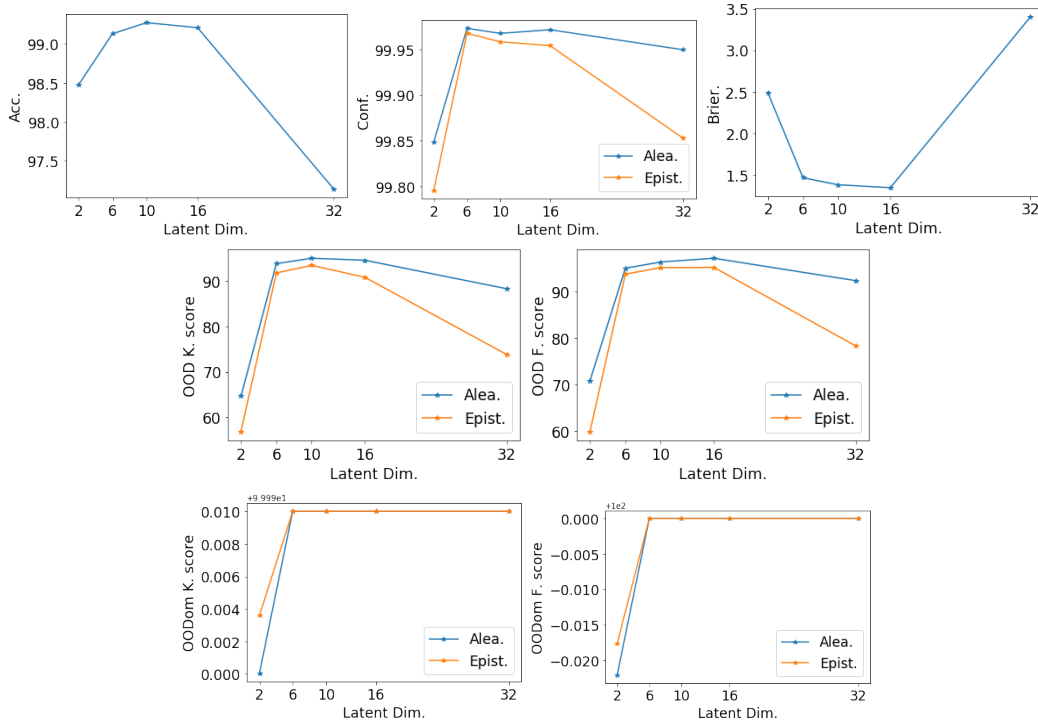
Figure 13: Accuracy and uncertainty scores of PostNet with latent dimension in $[2, 6, 10, 32]$ on the MNIST dataset. OOD and OODom scores are computed against scaled and unscaled KMNIST and FashionMNIST datasets. We observed that the performances remains high for medium dimensions (i.e. 6, 10) and drop for a too high dimension (i.e. 32).

	Acc.	Alea. Conf.	Epist. Conf.	Brier	OOD Alea.	OOD Epist.	OODom Alea.	OODom Epist.
Drop Out	71.73±0.2	92.18±0.1	84.38±0.3	49.76±0.2	72.94±0.3	41.68±0.5	28.3±1.8	47.1±3.3
Ensemble	*81.24±0.1	*96.61±0.0	93.25±0.1	38.88±0.1	*77.82±0.2	55.17±0.3	63.18±1.1	89.97±0.9
Distill.	72.11±0.4	91.72±0.2	90.73±0.2	88.04±0.1	**75.63±0.6**	52.18±2.1	17.76±0.0	17.76±0.0
KL-PN	48.84±0.5	78.01±0.6	77.99±0.7	83.11±0.6	59.32±1.1	58.03±0.8	17.79±0.0	20.25±0.2
RKL-PN	62.91±0.3	85.62±0.2	81.73±0.2	58.12±0.4	67.07±0.4	56.64±0.8	17.83±0.0	17.76±0.0
PostN Rad. (2)	76.43±0.1	94.59±0.1	94.02±0.1	37.59±0.3	72.91±0.4	69.26±1.1	99.99±0.0	*100.0±0.0
PostN Rad. (6)	76.46±0.3	94.75±0.1	*94.34±0.1	*37.39±0.4	72.83±0.6	*72.82±0.7	*100.0±0.0	*100.0±0.0
PostN Rad. (10)	75.43±0.2	94.16±0.1	93.64±0.1	39.3±0.4	71.94±0.3	70.99±0.5	*100.0±0.0	*100.0±0.0
PostN IAF (2)	76.75±0.2	**94.78±0.1**	92.98±0.2	37.87±0.5	73.07±0.5	65.61±1.0	*100.0±0.0	*100.0±0.0
PostN IAF (6)	**76.79±0.1**	94.73±0.0	93.7±0.1	37.86±0.2	73.58±0.2	69.74±0.3	*100.0±0.0	*100.0±0.0
PostN IAF (10)	75.92±0.2	94.48±0.1	93.23±0.2	39.09±0.3	72.4±0.3	69.04±0.3	*100.0±0.0	*100.0±0.0
PostN MoG (2)	44.7±5.9	54.12±7.9	52.12±7.8	68.57±5.2	48.53±3.6	47.45±4.1	99.91±0.0	99.96±0.0
PostN MoG (6)	71.05±1.6	91.21±1.0	86.91±1.2	46.37±2.0	73.49±0.6	56.04±3.8	98.04±0.7	99.62±0.1
PostN MoG (10)	71.63±1.3	91.57±0.8	88.92±0.8	46.07±1.9	72.61±0.3	56.28±1.8	99.88±0.0	*100.0±0.0

Table 12: Results on CIFAR10 dataset with all models with convolutional architecture. It shows results with different density types. Number into parentheses indicates flow size (for radial flow and IAF) or number of components (for MoG).

	Acc.	Alea. Conf.	Epist. Conf.	Brier	OOD S. Alea.	OOD S. Epist.	OODom S. Alea.	OODom S. Epist.
Ensemble	*91.34±0.0	*99.1±0.0	98.77±0.0	17.69±0.1	*80.1±0.3	75.14±0.2	21.1±3.1	24.42±3.7
RKL-PN	60.05±0.7	85.63±0.8	82.11±1.3	70.84±0.9	50.97±3.9	55.37±4.3	56.16±1.4	51.33±2.4
RKL-PN w/ C100	88.18±0.1	95.44±0.3	94.15±0.3	79.99±2.0	56.67±2.1	73.37±2.3	57.06±1.7	50.31±1.4
PostNet	**90.05±0.1**	**98.87±0.0**	*98.82±0.0	*15.44±0.1	76.04±0.4	*75.57±0.4	*87.65±0.3	*92.13±0.5

Table 13: Results with VGG16 on CIFAR10 on classic split (79%, 5%, 16%). RKL-PN w/ C100 uses CIFAR100 as training OOD.
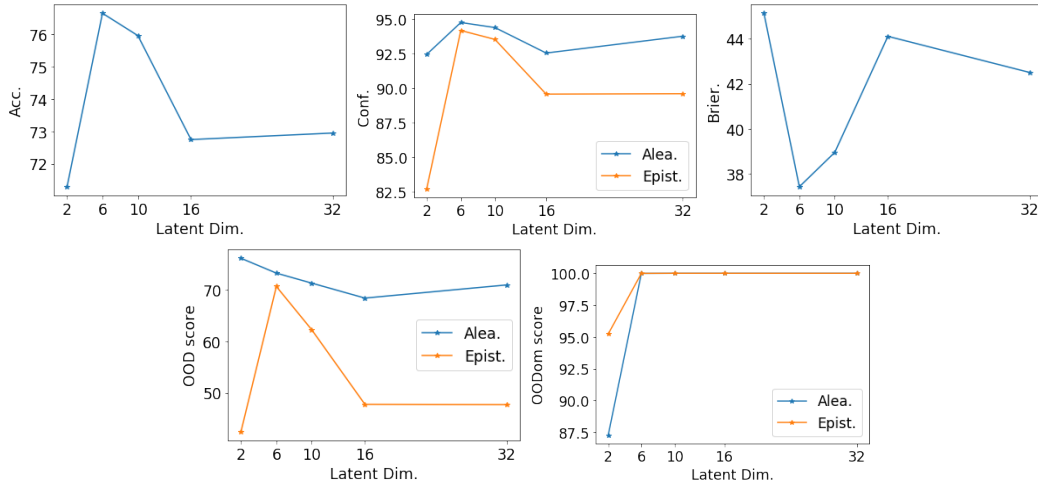
Figure 14: Accuracy and uncertainty scores of PostNet with latent dimension in $[2, 6, 10, 32]$ on the CIFAR10 dataset. OOD and OODom scores are computed against scaled and unscaled SVHN dataset. We observed that the performances remains high for medium dimensions (i.e. 6, 10) and drop for a too high dimension (i.e. 32).