# NATURAL POSTERIOR NETWORK: DEEP BAYESIAN PREDICTIVE UNCERTAINTY FOR EXPONENTIAL FAMILY DISTRIBUTIONS

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Uncertainty awareness is crucial to develop reliable machine learning models. In this work, we propose the Natural Posterior Network (NatPN) for fast and high-quality uncertainty estimation for any task where the target distribution belongs to the exponential family. Thus, NatPN finds application for both classification and general regression settings. Unlike many previous approaches, NatPN does not require out-of-distribution (OOD) data at training time. Instead, it leverages Normalizing Flows to fit a single density on a learned low-dimensional and task-dependent latent space. For any input sample, NatPN uses the predicted likelihood to perform a Bayesian update over the target distribution. Theoretically, NatPN assigns high uncertainty far away from training data. Empirically, our extensive experiments on calibration and OOD detection show that NatPN delivers highly competitive performance for classification, regression and count prediction tasks.

## 1 INTRODUCTION

Accurate and rigorous uncertainty estimation is key for reliable machine learning models in safety-critical domains [48]. It quantifies the confidence of machine learning models, thus allowing them to validate knowledgeable predictions corresponding to correct/wrong predictions, flag predictions on unknown input domains corresponding to anomaly or Out-of-Distribution detection, or detect natural shifts of the data facilitating real-time model maintenance (Filos et al., 2019; Malinin et al., 2021; Ovadia et al., 2019). Specifically, a reliable model can handle all these failure modes with high-quality estimates of *aleatoric* and *epistemic* uncertainty (Gal, 2016). These two levels of uncertainty allow a model to account for both irreducible data uncertainty (e.g. a fair dice's chance of $1/6$ for each face) and uncertainty due to the lack of knowledge about unseen data (e.g. input features differing significantly from training data or a covariate shift) respectively. Aleatoric and epistemic uncertainty levels can eventually be combined into an overall *predictive uncertainty* (Gal, 2016).

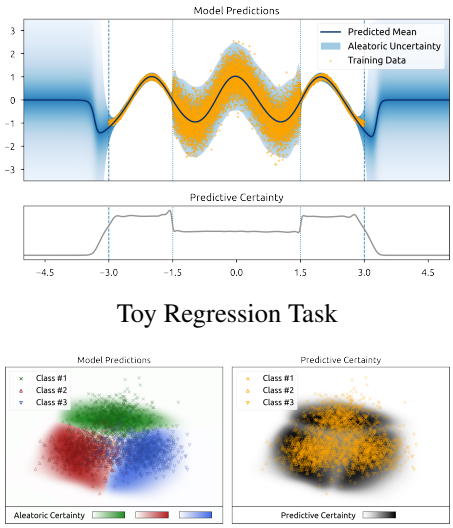


Figure 1: Visualization of the aleatoric and predictive uncertainty estimates of NatPN on two toy regressions and classification tasks. NatPN correctly assigns higher uncertainty to regions far from the training data.

Traditional neural networks are not readily applicable in safety-critical domains as they show overconfident prediction, in particular on data that is different from training data (Guo et al., 2017; Lakshminarayanan et al., 2017). To mitigate this problem, an important model family for uncertainty estimation directly predicts the parameters of a *conjugate prior distribution* on the predicted *target probability distribution*, thus accounting for the different levels of uncertainty. These models are efficient as they only require *a single forward*

*pass* for target and uncertainty prediction. Most of those models focus on classification and thus predict parameters of a Dirichlet distribution (Biloš et al., 2019; Charpentier et al., 2020; Gast & Roth, 2018; Malinin & Gales, 2018; 2019; Nandy et al., 2020; Sensoy et al., 2020; Shi et al., 2020; Zhao et al., 2020b). However, only two works Amini et al. (2020); Malinin et al. (2020a) have focused on regression by learning parameters of a Normal Inverse-Gamma (NIG) distribution as conjugate prior. Hence, all these models are limited to a single task (e.g. either classification or regression). Some approaches even require out-of-distribution (OOD) data at training time (Malinin & Gales, 2018; 2019) which is an unrealistic assumption in many applications where anomalies are a priori diverse, rare or unknown.

**Our contribution.** We propose Natural Posterior Network (NatPN) as a new approach parametrizing conjugate prior distributions for versatile uncertainty estimation. NatPN is motivated from both the theoretical and practical perspective. **(1)** NatPN can estimate predictive uncertainty for *any* task described by the general group of exponential family distributions contrary to existing approaches from this family of models. Notably, this encompasses very common tasks such as classification, regression and count prediction which can be described with Categorical, Normal and Poisson distributions, respectively. **(2)** In theory, NatPN is based on a *new unified exponential family framework* which performs an input-dependent Bayesian update. For every input, it predicts the parameters of the posterior over the target exponential family distribution. We show that this Bayesian update is *guaranteed* to predict high uncertainty far from training data. **(3)** In practice, NatPN requires *no OOD data for training*, only adds *a single normalizing flow* density to the last predictor layer and provides fast uncertainty estimation in a *single forward pass*. Our extensive experiments showcase the high performances of NatPN for various criteria (accuracy, calibration, OOD and shift detection) and tasks (classification, regression and count prediction). We illustrate the accurate aleatoric and predictive uncertainty predictions of NatPN on two toy examples for classification and regression in Fig. 1. *None of the conjugate prior related works* have similar theoretical and practical properties.

## 2 Related Work

In this section, we describe other work related to uncertainty estimation for supervised learning. We refer to Gawlikowski et al. (2021) for a detailed survey on uncertainty estimation in deep learning.

*Sampling-based methods:* A first family of models estimates uncertainty by aggregating statistics (e.g. mean and variance) from different samples of an implicit predictive distribution. Examples are ensemble (Kim & Ghahramani, 2012; Lakshminarayanan et al., 2017; Simpson et al., 2012; Wen et al., 2020; Wenzel et al., 2020) and dropout (Gal & Ghahramani, 2016) models which provide high-quality uncertainty estimates (Ovadia et al., 2019) at the cost of an expensive sampling phase at inference time. Moreover, ensembles usually require training multiple models. Further, Bayesian neural networks (BNN) (Blundell et al., 2015; Maddox et al., 2019; Ritter et al., 2018) model the uncertainty on the weights and also require multiple samples to estimate the uncertainty on the final prediction. While recent BNNs have shown reasonably good performances (Dusenberry et al., 2020; Farquhar et al., 2020; Osawa et al., 2019), modelling the distribution on the weights suffer from pathological behavior thus limiting these approaches in practice (Foong et al., 2020; Graves, 2011; Izmailov et al., 2021). In particular, Izmailov et al. (2021) uses an enormous computation budget by parallelizing the computation over 512 TPUv3 devices and running tens of thousands of training epochs to achieve a more exact Bayesian inference which is not suitable for practical applications. In contrast, NatPN predicts uncertainty in *a single forward pass* with a *closed-form posterior distribution* over the target variable. NatPN *does not* model uncertainty on the weights.

*Sampling-free methods:* A second family of models is capable of estimating uncertainty in a single forward pass. Beyond the family of models parametrizing conjugate prior distributions which is the main focus of this paper (Amini et al., 2020; Biloš et al., 2019; Charpentier et al., 2020; Gast & Roth, 2018; Malinin & Gales, 2018; 2019; Malinin et al., 2020a; Nandy et al., 2020; Sensoy et al., 2020; Shi et al., 2020; Zhao et al., 2020b), we differentiate between three other families of sampling-free models for uncertainty estimation. A family of models aims at learning deep Gaussian processes by learning random features projections or by learning inducing points (Biloš et al., 2019; Lakshminarayanan et al., 2020; van Amersfoort et al., 2020; 2021). Another family of models aims at propagating uncertainty across layers (Gast & Roth, 2018; Postels et al., 2019; Shekhovtsov & Flach, 2019; Wang et al., 2016). They model uncertainty at the activation and/or activation level and are

generally constrained to specific transformations. In contrast, NatPN only models the uncertainty on the predicted target variable and does not enforce any constraint on the encoder architecture. Further, some of the models propagating uncertainty already used the exponential family framework (Ranganath et al., 2015; Wang et al., 2016). However, while they parametrize exponential family distributions, NatPN parametrizes the *conjugate prior of the target exponential family distributions* which accounts for the epistemic uncertainty. Finally, while the family of calibration models aims at calibrating predictions (Kuleshov et al., 2018; Moon et al., 2020; Rahimi et al., 2020; Song et al., 2019; Zhao et al., 2020a), NatPN aims at accurately modelling both aleatoric and epistemic uncertainty on in- and out-of-distribution data.

## 3 Natural Posterior Network

At the very core of NatPN stands the Bayesian update rule: $\mathbb{Q}(\boldsymbol{\theta} \,|\, \mathcal{D}) \propto \mathbb{P}(\mathcal{D} \,|\, \boldsymbol{\theta}) \times \mathbb{Q}(\boldsymbol{\theta})$ where $\mathbb{P}(\mathcal{D} \,|\, \boldsymbol{\theta})$ is the target distribution of the target data $\mathcal{D}$ given its parameter $\boldsymbol{\theta}$, and $\mathbb{Q}(\boldsymbol{\theta})$ and $\mathbb{Q}(\boldsymbol{\theta} \,|\, \mathcal{D})$ are the prior and posterior distributions, respectively, over the target distribution parameters. The target distribution $\mathbb{P}(\mathcal{D} \,|\, \boldsymbol{\theta})$ could be any likelihood describing the observed target labels. The Bayesian update has three main advantages: **(1)** it introduces a prior belief which represents the safe default prediction if no data is observed, **(2)** it updates the prior prediction based on observed target labels, and **(3)** it assigns a confidence for the new target prediction given the aggregated evidence count of observed target labels. While NatPN is capable to perform a Bayesian update for every possible input given the observed training data, we first recall the Bayesian background for a single exponential family distribution.

| Likelihood $\mathbb{P}$ | Conjugate Prior $\mathbb{Q}$ | Parametrization Mapping $m$ | Bayesian Loss (Eq. 7) |
|---|---|---|---|
| $y \sim \mathrm{Cat}(\boldsymbol{p})$ | $\boldsymbol{p} \sim \mathrm{Dir}(\boldsymbol{\alpha})$ | $\boldsymbol{\chi} = \boldsymbol{\alpha}/n$<br>$n = \sum_c \alpha_c$ | **(1)** $= \psi(\alpha_{y*}^{(i)}) - \psi(\alpha_0^{(i)})$<br>**(2)** $= \log B(\boldsymbol{\alpha}^{(i)}) + (\alpha_0^{(i)} - C)\psi(\alpha_0^{(i)}) - \sum_c (\alpha_c^{(i)} - 1)\psi(\alpha_c^{(i)})$ |
| $y \sim \mathcal{N}(\mu, \sigma)$ | $\mu, \sigma \sim \mathcal{N}\Gamma^{-1}(\mu_0, \lambda, \alpha, \beta)$ | $\boldsymbol{\chi} = \begin{pmatrix} \mu_0 \\ \mu_0^2 + \frac{2\beta}{n} \end{pmatrix}$<br>$n = \lambda = 2\alpha$ | **(1)** $= \frac{1}{2}\left(-\frac{\alpha}{\beta}(y-\mu_0)^2 - \frac{1}{\lambda} + \psi(\alpha) - \log\beta - \log 2\pi\right)$<br>**(2)** $= \frac{1}{2} + \log\left((2\pi)^{\frac{1}{2}}\beta^{\frac{3}{2}}\Gamma(\alpha)\right) - \frac{1}{2}\log\lambda + \alpha - (\alpha + \frac{3}{2})\psi(\alpha)$ |
| $y \sim \mathrm{Poi}(\lambda)$ | $\lambda \sim \Gamma(\alpha, \beta)$ | $\chi = \alpha/n$<br>$n = \beta$ | **(1)** $= (\psi(\alpha) - \log\beta)y - \frac{\alpha}{\beta} - \sum_{k=1}^{y} \log k$<br>**(2)** $= \alpha + \log\Gamma(\alpha) - \log\beta + (1-\alpha)\psi(\alpha)$ |

Table 1: Examples of Exponential Family Distributions where $\psi(x)$ and $B(x)$ denote Digamma and Beta function, respectively.

**Exponential Family Distribution.** Distributions from the exponential family are very widely used and have favorable analytical properties. Indeed, **(1)** they cover a wide range of target variables like discrete, continuous, counts or spherical coordinates, and **(2)** they benefit from intuitive and generic formulae for their parameters, density functions and statistics which can often be evaluated in closed-form. Important examples of exponential family distributions are Normal, Categorical and Poisson distributions (see Tab. 1). Formally, an exponential family distribution on a target variable $y \in \mathbb{R}$ with *natural parameters* $\boldsymbol{\theta} \in \mathbb{R}^L$ can be denoted as

$$\mathbb{P}(y \,|\, \boldsymbol{\theta}) = h(y) \exp\left(\boldsymbol{\theta}^T \boldsymbol{u}(y) - A(\boldsymbol{\theta})\right) \tag{1}$$

where $h : \mathbb{R} \to \mathbb{R}$ is the *carrier or base measure*, $A : \mathbb{R}^L \to \mathbb{R}$ the *log-normalizer* and $\boldsymbol{u} : \mathbb{R} \to \mathbb{R}^L$ the *sufficient statistics* Bishop (2006); Nielsen & Nock (2010). The entropy of an exponential family distribution can always be written as $\mathbb{H}[\mathbb{P}] = A(\boldsymbol{\theta}) - \boldsymbol{\theta}^T \nabla_{\boldsymbol{\theta}} A(\boldsymbol{\theta}) - \mathbb{E}[\log h(y)]$ Nielsen & Nock (2010). An exponential family distribution always admits a conjugate prior, which often also is a member of the exponential family:

$$\mathbb{Q}(\boldsymbol{\theta} \,|\, \boldsymbol{\chi}, n) = \eta(\boldsymbol{\chi}, n) \exp\left(n\, \boldsymbol{\theta}^T \boldsymbol{\chi} - nA(\boldsymbol{\theta})\right) \tag{2}$$

where $\eta(\boldsymbol{\chi}, n)$ is a normalization coefficient, $\boldsymbol{\chi} \in \mathbb{R}^L$ are *prior parameters* and $n \in \mathbb{R}^+$ is the *evidence*. Given a set of $N$ target observations $\{y^{(i)}\}_i^N$, it is easy to compute a closed-form Bayesian update $\mathbb{Q}(\boldsymbol{\theta} \,|\, \boldsymbol{\chi}^{\text{post}}, n^{\text{post}}) \propto \mathbb{P}(\{y^{(i)}\}_i^N \,|\, \boldsymbol{\theta}) \times \mathbb{Q}(\boldsymbol{\theta} \,|\, \chi^{\text{prior}}, n^{\text{prior}})$:

$$\mathbb{Q}(\boldsymbol{\theta} \,|\, \boldsymbol{\chi}^{\text{post}}, n^{\text{post}}) \propto \exp\left(n^{\text{post}} \boldsymbol{\theta}^T \boldsymbol{\chi}^{\text{post}} - n^{\text{post}} A(\boldsymbol{\theta})\right) \tag{3}$$

where $\boldsymbol{\chi}^{\text{post}} = \frac{n^{\text{prior}}\boldsymbol{\chi}^{\text{prior}} + \sum_j^N \boldsymbol{u}(y^{(j)})}{n^{\text{prior}} + N}$ and $n^{\text{post}} = n^{\text{prior}} + N$. We see that $\boldsymbol{\chi}^{\text{prior}}$ (resp. $\boldsymbol{\chi}^{\text{post}}$) can be viewed as the average sufficient statistics of $n^{\text{prior}}$ (resp. $n^{\text{post}}$) fictitious samples Bishop (2006).

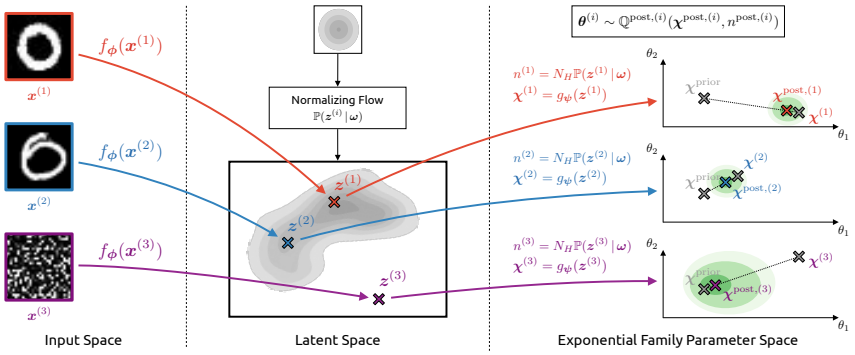


Figure 2: Overview of Natural Posterior Network. Inputs $\boldsymbol{x}^{(i)}$ are first mapped to a low-dimensional latent representation $\boldsymbol{z}^{(i)}$ by the encoder $f_{\boldsymbol{\phi}}$. From $\boldsymbol{z}^{(i)}$, the decoder $g_{\boldsymbol{\psi}}$ derives the parameter update $\boldsymbol{\chi}^{(i)}$ while a normalizing flow $\mathbb{P}_{\boldsymbol{\omega}}$ yields the evidence update $n^{(i)}$. Posterior parameters are obtained from a weighted combination of prior and update parameters according to $n^{\text{post},(i)}$.

Further, the average sufficient statistic of fictitious samples is equal to the expected sufficient statistic of the conjugate distribution, i.e. $\boldsymbol{\chi} = \mathbb{E}_{\mathbb{Q}(\boldsymbol{\chi},n)}[\boldsymbol{\theta}]$ (Brown, 1986; Diaconis P., 1979). Therefore, the parameter $\boldsymbol{\chi}^{\text{post}}$ carries the inherent aleatoric uncertainty on the target distribution with natural parameters $\boldsymbol{\theta}$, while the evidence $n^{\text{post}}$ aligns well with the epistemic uncertainty (i.e. a low evidence means few prior target observations). We emphasize that the natural conjugate prior parametrization $\boldsymbol{\chi}, n$ is often different from the "well-known" parametrization $\boldsymbol{\kappa}$ used by standard coding libraries. By definition, a bijective mapping $m(\boldsymbol{\kappa}) = (\boldsymbol{\chi}, n)$ from the natural parametrization to the commonly used parametrization always exists (see examples in Tab. 1). Finally, exponential family distributions always admit a closed-form predictive posterior distribution (Gelman et al., 2004).

**Input-Dependent Bayesian Update for Exponential Family Distributions.** We propose to leverage the power of exponential family distributions for the more complex task when the prediction $y^{(i)}$ depends on the input $\boldsymbol{x}^{(i)}$. Hence, NatPN extends the Bayesian treatment of a single exponential family distribution prediction by predicting an individual posterior update per input. We distinguish between the chosen prior parameters $\boldsymbol{\chi}^{\text{prior}}$, $n^{\text{prior}}$ shared among samples, and the additional predicted parameters $\boldsymbol{\chi}^{(i)}$, $n^{(i)}$ dependent on the input $\boldsymbol{x}^{(i)}$ leading to the updated posterior parameters:

$$\boldsymbol{\chi}^{\text{post},(i)} = \frac{n^{\text{prior}}\boldsymbol{\chi}^{\text{prior}} + n^{(i)}\boldsymbol{\chi}^{(i)}}{n^{\text{prior}} + n^{(i)}}, \qquad n^{\text{post},(i)} = n^{\text{prior}} + n^{(i)} \tag{4}$$

Equivalently, NatPN may be interpreted as predicting a set of $n^{(i)}$ pseudo observations $\{y^{(j)}\}_j^{(i)}$ such that their aggregated sufficient statistics satisfy $\sum_j^{n^{(i)}} y^{(j)} = n^{(i)}\boldsymbol{\chi}^{(i)}$, and perform the respective Bayesian update. This Bayesian update works for *any* choice of exponential family distributions as long as parameters are mapped to their standard form (see Tab. 1). According to the *principle of maximum entropy* (Rosenkrantz, 1989), a practical choice for the prior is to enforce high entropy for the prior distribution which is usually considered less informative. It is typically achieved when the prior pseudo-count $n^{\text{prior}}$ is small and the prior parameter $\boldsymbol{\chi}^{\text{prior}}$ shows a high aleatoric uncertainty.

Hence, NatPN proposes a generic way to perform the input-dependent Bayesian update $\boldsymbol{\chi}^{(i)}$, $n^{(i)}$ for *any* exponential family distribution in three steps (see Fig. 2): **(1)** An encoder $f_{\boldsymbol{\phi}}$ maps the input $\boldsymbol{x}^{(i)}$ onto a low-dimensional latent vector $\boldsymbol{z}^{(i)} = f_{\phi}(\boldsymbol{x}^{(i)}) \in \mathbb{R}^H$ representing useful features for the prediction task (see left Fig.2). Note that the architecture of the encoder can be arbitrarily complex. Then, **(2)** the latent representation $\boldsymbol{z}^{(i)}$ is used in two different ways to predict the parameter update $\boldsymbol{\chi}^{(i)}$ and the evidence update $n^{(i)}$ (see center Fig.2). On the one hand, a linear decoder $g_{\boldsymbol{\psi}}$ is trained to output the parameter update $\boldsymbol{\chi}^{(i)} = g_{\boldsymbol{\psi}}(\boldsymbol{z}^{(i)}) \in \mathbb{R}^L$ accounting for the aleatoric uncertainty. On the other hand, a single normalized density is trained to output the evidence update $n^{(i)} = N_H\, \mathbb{P}(\boldsymbol{z}^{(i)} \,|\, \boldsymbol{\omega})$ accounting for the epistemic uncertainty. The intuition is that increasing the evidence on training data during training forces the evidence everywhere else (incl. far from training data) to decrease thanks to the density normalization constraint. The constant $N_H$ is a certainty budget distributed by the normalized density $\mathbb{P}(\boldsymbol{z}^{(i)} \,|\, \boldsymbol{\omega})$ over the latent representations $\boldsymbol{z}^{(i)}$ i.e. $N_H = \int N_H\, \mathbb{P}(\boldsymbol{z}^{(i)} \,|\, \boldsymbol{\omega}) d\boldsymbol{z}^{(i)} = \int n^{(i)} d\boldsymbol{z}^{(i)}$. In practice, we observed that scaling the certainty budget w.r.t. the latent dimension $H$ helped the density to cover larger volumes in higher dimension

(see app.). Finally, **(3)** NatPN computes the posterior parameters $\boldsymbol{\chi}^{\text{post},(i)}$ and $n^{\text{post},(i)}$ which can be viewed respectively as the mean and concentration of the posterior distribution (see right Fig.2). Note that the posterior parameter $\boldsymbol{\chi}^{\text{post},(i)}$ is a simple weighted average of the prior parameter $\boldsymbol{\chi}^{\text{prior}}$ and the update parameter $\boldsymbol{\chi}^{(i)}$ as shown by Eq. 4.

NatPN extends PostNet (Charpentier et al., 2020) which also performs an input-dependent Bayesian update with density estimation. Yet, it has three crucial differences which lead to major practical improvements. First, the new exponential family framework is significantly more flexible and is not restricted to classification. Second, the Dirichlet $\boldsymbol{\alpha}$ parameter computation is different: NatPN computes the $\boldsymbol{\chi}$ parameters – which can be viewed as standard softmax output – and the $n$ evidence separately (i.e. $\boldsymbol{\alpha} = n\boldsymbol{\chi}$) while PostNet computes one evidence pseudo-count per class. Third, NatPN is computationally more efficient. It requires a single density while PostNet requires $C$ densities.

**ID and OOD Uncertainty Estimates.** NatPN intuitively leads to reasonable uncertainty estimation for the two limit cases of strong ID and OOD inputs (see red and purple samples in Fig. 2). For very likely *in-distribution* data (i.e. $\mathbb{P}(\boldsymbol{z}^{(i)} \mid \boldsymbol{\omega}) \to \infty$), the posterior parameter overrules the prior (i.e. $\boldsymbol{\chi}^{\text{post},(i)} \to \boldsymbol{\chi}^{(i)}$). Conversely, for very unlikely *out-of-distribution* data (i.e. $\mathbb{P}(\boldsymbol{z}^{(i)} \mid \boldsymbol{\omega}) \to 0$), the prior parameter takes over in the posterior update (i.e. $\boldsymbol{\chi}^{\text{post},(i)} \to \boldsymbol{\chi}^{\text{prior}}$). Hence, the choice of the prior parameter should reflect the default prediction when the model lacks knowledge. We formally show under mild assumptions on the encoder that NatPN predicts very low additional evidence ($n^{(i)} \approx 0$) for (almost) any input $\boldsymbol{x}^{(i)}$ far away from the training data (i.e. $||\boldsymbol{x}^{(i)}|| \to +\infty$), thus recovering prior predictions (i.e. $\boldsymbol{\chi}^{\text{post},(i)} \approx \boldsymbol{\chi}^{\text{prior}}$) (see proof in app.).

**Theorem 1.** *Let a NatPN model be parametrized with a (deep) encoder $f_\phi$ with ReLU activations, a decoder $g_\psi$ and the density $\mathbb{P}(\boldsymbol{z} \mid \boldsymbol{\omega})$. Let $f_\phi(\boldsymbol{x}) = V^{(l)}\boldsymbol{x} + a^{(l)}$ be the piecewise affine representation of the ReLU network $f_\phi$ on the finite number of affine regions $Q^{(l)}$ (Arora et al., 2018). Suppose that $V^{(l)}$ have independent rows and the density function $\mathbb{P}(\boldsymbol{z} \mid \boldsymbol{\omega})$ has bounded derivatives, then for almost any $\boldsymbol{x}$ we have $\mathbb{P}(f_\phi(\delta \cdot \boldsymbol{x}) \mid \boldsymbol{\omega}) \underset{\delta \to \infty}{\to} 0$. i.e the evidence becomes small far from training data.*

This theorem only requires that the density avoids very unlikely pathological behavior with unbounded derivatives (Dix, 2013). A slightly weaker conclusion holds using the notion of limit in density if the density function does not have bounded derivatives (Niculescu & Popovici, 2011). Finally, the independent rows condition is realistic for trained networks with no constant output (Hein et al., 2019). It advantageously leads NatPN to consistent uncertainty estimation contrary to standard ReLU networks which are overconfident far from training data (Hein et al., 2019).

**Bayesian NatPN Ensemble.** Interestingly, it is natural to extend the Bayesian treatment of a single NatPN to an ensemble of NatPN models. An ensemble of $m$ NatPN models is intuitively equivalent to performing $m$ successive Bayesian updates using each NatPN member all separately. More formally, given an input $\boldsymbol{x}^{(i)}$ and an ensemble of $m$ jointly trained NatPN models, the Bayesian update for the posterior distribution becomes $\boldsymbol{\chi}^{\text{post},(i)} = \frac{n^{\text{prior}}\boldsymbol{\chi}^{\text{prior}} + \sum_k^m n_k^{(i)}\boldsymbol{\chi}_k^{(i)}}{n^{\text{prior}} + \sum_k^m n_k^{(i)}}$ and $n^{\text{post},(i)} = n^{\text{prior}} + \sum_k^m n_k^{(i)}$. Note that the standard Bayesian averaging which is used in many ensembling methods (Chipman et al., 2007; Lakshminarayanan et al., 2017; Wen et al., 2020; Wenzel et al., 2020) is different from this Bayesian combination. While Bayesian averaging assume that only one model is correct, the Bayesian combination of NatPN allows *more* or *none* of the models to be "expert" for some input (Monteith et al., 2011). For example, an input $\boldsymbol{x}^{(i)}$ unfamiliar to every model $m$ (i.e. $n_m^{(i)} \approx 0$) would recover the prior default prediction $\boldsymbol{\chi}^{\text{prior}}, n^{\text{prior}}$. Existing models already had similar properties for Bayesian combination of classifiers (Kim & Ghahramani, 2012; Simpson et al., 2012).

### 3.1 Optimization

The choice of the optimization procedure is of primary importance in order to obtain both high-quality target predictions and uncertainty estimates regardless of the task.

**Bayesian Loss.** We follow Charpentier et al. (2020) and aim at minimizing the principled Bayesian loss:

$$\mathcal{L}^{(i)} = -\underbrace{\mathbb{E}_{\boldsymbol{\theta}^{(i)} \sim \mathbb{Q}^{\text{post},(i)}}[\log \mathbb{P}(y^{(i)} \mid \boldsymbol{\theta}^{(i)})]}_{(1)} - \underbrace{\mathbb{H}[\mathbb{Q}^{\text{post},(i)}]}_{(2)} \tag{5}$$

where $\mathbb{H}[\mathbb{Q}^{\text{post},(i)}]$ denotes the entropy of the predicted posterior distribution $\mathbb{Q}^{\text{post},(i)}$. Similarly to the ELBO loss, this loss is guaranteed to be optimal when the predicted posterior distribution is close to the true posterior distribution $\mathbb{Q}^*(\boldsymbol{\theta} \,|\, \boldsymbol{x}^{(i)})$ i.e. $\mathbb{Q}^{\text{post},(i)} \approx \mathbb{Q}^*(\boldsymbol{\theta} \,|\, \boldsymbol{x}^{(i)})$ (Bissiri et al., 2016; Shawe-Taylor & Williamson, 1997; Zellner, 1988). However, this loss is generally *not* equal to the ELBO loss especially for real valued targets i.e. $y \in \mathbb{R}$ (see app.). The term **(1)** is the expected likelihood under the predicted posterior distribution. It can be viewed as the Uncertain Cross Entropy (UCE) loss (Biloš et al., 2019) which is known to reduce uncertainty on observed data. The term **(2)** is an entropy regularizer acting as a prior which favors uninformative distributions $\mathbb{Q}^{\text{post},(i)}$ with high entropy. In our case, we assume the likelihood $\mathbb{P}(y^{(i)} \,|\, \boldsymbol{\theta}^{(i)})$ and the posterior $\mathbb{Q}^{\text{post},(i)}$ to be members of the exponential family. We take advantage of the convenient computations for such distributions and derive a more explicit formula for the Bayesian loss of Eq. 7 (see derivation in the appendix):

$$\mathcal{L}^{(i)} \propto \mathbb{E}[\boldsymbol{\theta}]^T \boldsymbol{u}(y^{(i)}) - \mathbb{E}[A(\boldsymbol{\theta})] - \lambda \, \mathbb{H}[\mathbb{Q}^{\text{post},(i)}] \tag{6}$$

where $\lambda$ is an additional regularization weight tuned with a grid search. Note that the term $\mathbb{E}[\boldsymbol{\theta}]^T \boldsymbol{u}(y^{(i)})$ favors a good alignment of the expected sufficient statistic $\mathbb{E}[\boldsymbol{\theta}] = \boldsymbol{\chi}$ with the observed sufficient statistic $\boldsymbol{u}(y^{(i)})$. In practice, all terms can be computed efficiently in closed form for most exponential family distributions (see examples in Tab. 1). In particular, simplifications are possible when the conjugate prior distribution is also in an exponential family which is often the case. Ultimately Eq. (6) applies to *any* exponential family distribution unlike Charpentier et al. (2020).

**Optimization Scheme.** NatPN is fully differentiable using the closed-form Bayesian loss. Thus, we train the encoder $f_{\boldsymbol{\phi}}$, the parameter decoder $g_{\boldsymbol{\psi}}$ and the normalizing flow $\mathbb{P}(\boldsymbol{z}^{(i)} \,|\, \boldsymbol{\omega})$ w.r.t. parameters $\boldsymbol{\phi}, \boldsymbol{\psi}, \boldsymbol{\omega}$ jointly. Additionally, we observed that "warm-up" training and "fine-tuning" of the density helped to improve uncertainty estimation for more complex flows and datasets. Thus, we trained the normalizing flow density to maximize the likelihood of the latent representations before and after the joint optimization while keeping all other parameters fixed.

### 3.2 Model Limitations

**Task-Specific OOD.** It has been shown that density estimation is unsuitable for acting on the raw image input (Choi et al., 2019; Nalisnick et al., 2019; 2020) or on a non-carefully transformed space (Lan & Dinh, 2020). To circumvent this issue, NatPN does not perform OOD detection directly on the input but rather fits a normalizing flow on a learned space. In particular, the latent space is **(1)** low-dimensional, **(2)** task-specific and **(3)** encodes meaningful semantic features. Similarly, Charpentier et al. (2020); Kirichenko et al. (2020); Morningstar et al. (2020); Winkens et al. (2020) already improved OOD detection of density-based methods by leveraging a task-induced bias or low-dimensional statistics. In the case of NatPN, the low-dimensional latent space has to contain relevant features to linearly predict the sufficient statistics required for the task. For example, NatPN aims at a linearly separable latent space for classification. The downside is that NatPN is capable of detecting OOD samples only with respect to the considered task and requires labeled examples during training. As an example, NatPN likely fails to detect a change of image color if the task aims at classifying object shapes and the latent space has no notion of color. Hence, we underline that NatPN comes with a task-dependent OOD definition, which is a reasonable choice in practice.

**Model-Task Mismatch.** Second, we emphasize that the uncertainty estimation quality of NatPN for (close to) ID data depends on the convergence of the model, the encoder architecture (e.g. MLP, Conv., DenseDepth (Eigen et al., 2014)) and the target distribution (e.g. Poisson, Normal distributions) choice which should match the task needs. However, we show *empirically* that NatPN provides high quality uncertainty estimates in practice on a wide range of tasks. Further, we show *theoretically* that NatPN leads to uncertain prediction far away from training data for *any* exponential family target distributions. In comparison, Meinke & Hein (2020) showed akin guarantees for classification only.

## 4 Experiments

In this section, we compare NatPN to existing methods on extensive experiments including three different tasks: classification, regression and count prediction. For each task type, we evaluate the prediction quality based on target error and uncertainty metrics. These various set-ups aim at proving the versatility of NatPN. In particular, NatPN is the only model that adapts to all tasks and achieves high performances for all metrics without requiring multiple forward passes.

| | Accuracy | Brier | 9/10 Alea. | 9/10 Epist. | OODom Alea. | OODom Epist. |
|---|---|---|---|---|---|---|
| **Dropout** | $98.62 \pm 0.11$ | $3.79 \pm 0.29$ | $30.20 \pm 0.85$ | $32.57 \pm 1.45$ | $27.03 \pm 0.51$ | $95.30 \pm 1.66$ |
| **Ensemble** | $98.83 \pm 0.17$ | $3.00 \pm 0.54$ | $30.79 \pm 0.74$ | $32.61 \pm 1.06$ | $27.16 \pm 0.59$ | $99.97 \pm 0.01$ |
| **NatPE** | *$99.66 \pm 0.03$ | *$0.68 \pm 0.05$ | $77.05 \pm 1.93$ | $83.73 \pm 1.89$ | $99.99 \pm 0.00$ | *$100.00 \pm 0.00$ |
| **R-PriorNet** | $98.85 \pm 0.25$ | $2.01 \pm 0.47$ | $40.13 \pm 2.99$ | $30.07 \pm 0.81$ | ***$\mathbf{100.00 \pm 0.00}$** | $23.59 \pm 0.00$ |
| **EnD$^2$** | $93.95 \pm 2.35$ | $28.09 \pm 6.40$ | $26.35 \pm 0.60$ | $24.85 \pm 0.43$ | $84.43 \pm 15.21$ | $23.58 \pm 0.00$ |
| **PostNet** | $\mathbf{99.64 \pm 0.02}$ | $\mathbf{0.75 \pm 0.08}$ | $80.60 \pm 1.68$ | ***$\mathbf{92.57 \pm 1.41}$** | ***$\mathbf{100.00 \pm 0.00}$** | ***$\mathbf{100.00 \pm 0.00}$** |
| **NatPN** | $99.61 \pm 0.05$ | $1.04 \pm 0.29$ | ***$\mathbf{81.43 \pm 1.89}$** | $79.54 \pm 2.62$ | $99.98 \pm 0.00$ | ***$\mathbf{100.00 \pm 0.00}$** |

Table 2: Classification results on Sensorless Drive with Categorical target distribution. Best scores among all single-pass models are in bold. Best scores among all models are starred.

| | Accuracy | Brier | SVHN Alea. | SVHN Epist. | CelebA Alea. | CelebA Epist. | OODom Alea. | OODom Epist. |
|---|---|---|---|---|---|---|---|---|
| **Dropout** | $88.15 \pm 0.20$ | $19.59 \pm 0.41$ | $80.63 \pm 1.59$ | $73.09 \pm 1.51$ | $71.84 \pm 4.28$ | $71.04 \pm 3.92$ | $18.42 \pm 1.11$ | $49.69 \pm 9.10$ |
| **Ensemble** | *$89.95 \pm 0.11$ | $17.33 \pm 0.17$ | $85.26 \pm 0.84$ | $82.51 \pm 0.63$ | $76.20 \pm 0.87$ | $74.23 \pm 0.78$ | $25.30 \pm 4.02$ | $89.21 \pm 7.55$ |
| **NatPE** | $89.21 \pm 0.09$ | $17.41 \pm 0.12$ | $85.66 \pm 0.34$ | *$83.16 \pm 0.67$ | *$78.95 \pm 1.15$ | *$82.06 \pm 1.30$ | $87.27 \pm 1.79$ | *$98.88 \pm 0.26$ |
| **R-PriorNet** | $\mathbf{88.94 \pm 0.23}$ | ***$\mathbf{15.99 \pm 0.32}$** | $99.87 \pm 0.02$ | $99.94 \pm 0.01$ | $67.74 \pm 4.86$ | $59.55 \pm 7.90$ | $42.21 \pm 8.77$ | $38.25 \pm 9.82$ |
| **EnD$^2$** | $84.03 \pm 0.25$ | $40.84 \pm 0.36$ | ***$\mathbf{86.47 \pm 0.66}$** | $\mathbf{81.84 \pm 0.92}$ | $75.54 \pm 1.79$ | $75.94 \pm 1.82$ | $42.19 \pm 8.77$ | $15.79 \pm 0.27$ |
| **PostNet** | $87.95 \pm 0.20$ | $20.19 \pm 0.40$ | $82.35 \pm 0.68$ | $79.24 \pm 1.49$ | $72.96 \pm 2.33$ | $75.84 \pm 1.61$ | $85.89 \pm 4.10$ | $92.30 \pm 2.18$ |
| **NatPN** | $87.90 \pm 0.16$ | $19.99 \pm 0.46$ | $82.29 \pm 1.11$ | $77.83 \pm 1.22$ | $\mathbf{76.01 \pm 1.18}$ | $\mathbf{76.87 \pm 3.38}$ | ***$\mathbf{93.67 \pm 3.03}$** | $\mathbf{94.90 \pm 3.09}$ |

Table 3: Classification results on CIFAR-10 with Categorical target distribution. Best scores among all single-pass models are in bold. Best scores among all models are starred. Gray numbers indicate that R-PriorNet has seen samples from the SVHN dataset during training.

### 4.1 SETUP

In our experiments, we change the encoder architecture of **NatPN** to match the dataset needs. We perform a grid search over normalizing flows types (i.e. radial flows (Rezende & Mohamed, 2015) and MAF (Germain et al., 2015; Papamakarios et al., 2017)) and latent dimensions. We show further experiments on architecture, latent dimension, normalizing flow choices and certainty budget choice in the appendix. Furthermore, we use approximations of the log-Gamma $\log \Gamma(x)$ and the Digamma $\psi(x)$ functions for large input values to avoid unstable floating computations (see app.). As prior parameters, we set $\boldsymbol{\chi}^{\text{prior}} = \mathbf{1}_C / C, n^{\text{prior}} = C$ for classification, $\boldsymbol{\chi}^{\text{prior}} = (0, 100)^T, n^{\text{prior}} = 1$ for regression and $\chi^{\text{prior}} = 1, n^{\text{prior}} = 1$ for count prediction enforcing high entropy for priors distributions.

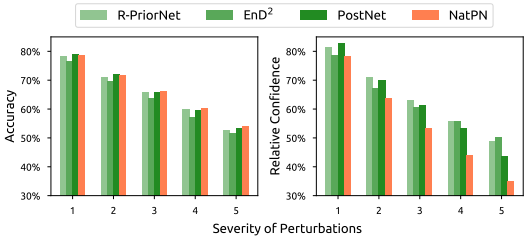


Figure 3: Averaged accuracy and confidence under 15 dataset shifts on CIFAR-10 (Hendrycks & Dietterich, 2019). On more severe perturbations (i.e. data further away from data distribution), NatPN maintains a competitive accuracy while assigning higher epistemic uncertainty as desired. Baselines provide a slower relative confidence decay.

**Baselines.** We focus on recent models parametrizing prior distributions over the target distribution. For classification, we compare NatPN to Reverse KL divergence Prior Networks **(R-PriorNet)** (Malinin & Gales, 2019), Ensemble Distribution Distillation **(EnD$^2$)** (Malinin et al., 2020b) and Posterior Networks **(PostNet)** (Charpentier et al., 2020). Note that Prior Networks require OOD training data — we use an auxiliary dataset when available and Gaussian noise otherwise. For regression, we compare to Evidential Regression **(EvReg)** (Amini et al., 2020). Beyond baselines parametrizing conjugate prior distributions, we also compare to dropout **(Dropout)** (Gal & Ghahramani, 2016) and ensemble **(Ensemble)** (Lakshminarayanan et al., 2017) models for all tasks. As suggested in (Ovadia et al., 2019), we use 5 samples for the two sampling baselines. Note that, unlike the other baselines or NatPN, the sampling baselines require multiple forward passes for uncertainty estimation. In all experiments, all models share the same backbone architecture (MLP, Conv. (Yamashita et al., 2018), DenseDepth (Eigen et al., 2014)). We use early stopping and perform a grid search on the hyperparameters of all models including learning rate, dropout rate and regularizing factor when applicable. We report average results over 5 initialization seeds with standard error of the mean (except for the larger dataset NYU Depth v2 dataset). Further model details are given in the appendix.

**Datasets.** We split all datasets into train, validation and test sets. For classification, we use one tabular dataset (**Sensorless Drive** (Dua & Graff, 2017)) and three image datasets (**MNIST** (LeCun et al., 2010), **FMNIST** (Xiao et al., 2017) and **CIFAR-10** (Krizhevsky et al., 2009)). For count prediction, we use the **Bike Sharing** dataset (Fanaee-T & Gama, 2014) to predict the number of bike rentals within an hour. For regression, we also use the Bike Sharing dataset where the target is viewed as continuous, real-world **UCI datasets** used in Amini et al. (2020); Hernandez-Lobato & Adams

| | RMSE | Calibration | Winter Epist. | Spring Epist. | Autumn Epist. | OODom Epist. |
|---|---|---|---|---|---|---|
| **Dropout-$\mathcal{N}$** | $70.20 \pm 1.30$ | $6.05 \pm 0.77$ | $15.26 \pm 0.51$ | $13.66 \pm 0.16$ | $15.11 \pm 0.46$ | $99.99 \pm 0.01$ |
| **Ensemble-$\mathcal{N}$** | $*48.02 \pm 2.78$ | $5.88 \pm 1.00$ | $42.46 \pm 2.29$ | $21.28 \pm 0.38$ | $21.97 \pm 0.58$ | $*100.00 \pm 0.00$ |
| **EvReg-$\mathcal{N}$** | $\mathbf{49.58 \pm 1.51}$ | $3.77 \pm 0.81$ | $17.19 \pm 0.76$ | $15.54 \pm 0.65$ | $14.75 \pm 0.29$ | $34.99 \pm 17.02$ |
| **NatPN-$\mathcal{N}$** | $49.85 \pm 1.38$ | $\mathbf{*1.95 \pm 0.34}$ | $\mathbf{*55.04 \pm 6.81}$ | $\mathbf{*23.25 \pm 1.20}$ | $\mathbf{*27.78 \pm 2.47}$ | $\mathbf{*100.00 \pm 0.00}$ |
| **Dropout-Poi** | $66.57 \pm 4.61$ | $55.00 \pm 0.22$ | $16.02 \pm 0.48$ | $13.48 \pm 0.38$ | $18.09 \pm 0.82$ | $\mathbf{*100.00 \pm 0.00}$ |
| **Ensemble-Poi** | $\mathbf{*48.22 \pm 2.06}$ | $55.31 \pm 0.21$ | $83.88 \pm 1.22$ | $34.21 \pm 1.81$ | $41.29 \pm 3.23$ | $\mathbf{*100.00 \pm 0.00}$ |
| **NatPN-Poi** | $51.79 \pm 0.78$ | $\mathbf{*31.04 \pm 1.81}$ | $\mathbf{*85.15 \pm 3.61}$ | $\mathbf{*37.03 \pm 2.35}$ | $\mathbf{*42.73 \pm 4.38}$ | $\mathbf{*100.00 \pm 0.00}$ |

Table 4: Results on the Bike Sharing Dataset with Normal $\mathcal{N}$ and Poison Poi target distributions. Best scores among all single-pass models are in bold. Best scores among all models are starred.

| | RMSE | Calibration | Energy Alea. | Energy Epist. | Concrete Alea. | Concrete Epist. | Kin8nm Alea. | Kin8nm Epist. |
|---|---|---|---|---|---|---|---|---|
| **Dropout** | $0.09 \pm 0.00$ | $3.13 \pm 0.43$ | $90.18 \pm 6.00$ | $99.94 \pm 0.06$ | $*100.00 \pm 0.00$ | $*100.00 \pm 0.00$ | **in-distribution** | |
| **Ensemble** | $*0.07 \pm 0.00$ | $2.69 \pm 0.49$ | $*100.00 \pm 0.00$ | $*100.00 \pm 0.00$ | $*100.00 \pm 0.00$ | $*100.00 \pm 0.00$ | | |
| **NatPE** | $0.08 \pm 0.00$ | $5.49 \pm 0.30$ | $*100.00 \pm 0.00$ | $*100.00 \pm 0.00$ | $*100.00 \pm 0.00$ | $*100.00 \pm 0.00$ | | |
| **EvReg** | $0.09 \pm 0.00$ | $3.74 \pm 0.53$ | $88.06 \pm 11.94$ | $88.06 \pm 11.94$ | $\mathbf{*100.00 \pm 0.00}$ | $86.84 \pm 13.16$ | **in-distribution** | |
| **NatPN** | $\mathbf{0.08 \pm 0.00}$ | $\mathbf{*2.04 \pm 0.45}$ | $\mathbf{*100.00 \pm 0.00}$ | $\mathbf{*100.00 \pm 0.00}$ | $\mathbf{*100.00 \pm 0.00}$ | $\mathbf{*100.00 \pm 0.00}$ | | |
| **Dropout** | $5.67 \pm 0.07$ | $*3.03 \pm 0.40$ | $9.33 \pm 0.36$ | $93.53 \pm 2.41$ | **in-distribution** | | $1.09 \pm 0.13$ | $64.30 \pm 7.14$ |
| **Ensemble** | $5.69 \pm 0.20$ | $3.81 \pm 0.67$ | $54.19 \pm 18.93$ | $*100.00 \pm 0.00$ | | | $72.57 \pm 19.32$ | $*100.00 \pm 0.00$ |
| **NatPE** | $*4.78 \pm 0.20$ | $5.58 \pm 1.27$ | $*100.00 \pm 0.00$ | $*100.00 \pm 0.00$ | | | $*100.00 \pm 0.00$ | $*100.00 \pm 0.00$ |
| **EvReg** | $6.04 \pm 0.18$ | $7.36 \pm 1.04$ | $8.93 \pm 0.02$ | $51.39 \pm 18.56$ | **in-distribution** | | $0.93 \pm 0.00$ | $34.44 \pm 20.95$ |
| **NatPN** | $\mathbf{5.83 \pm 0.23}$ | $\mathbf{5.41 \pm 1.33}$ | $\mathbf{*100.00 \pm 0.00}$ | $\mathbf{*100.00 \pm 0.00}$ | | | $\mathbf{*100.00 \pm 0.00}$ | $\mathbf{*100.00 \pm 0.00}$ |

Table 5: Regression results on models trained on different UCI datasets with Normal target distribution. The upper half displays models trained on Kin8nm, the lower half shows models trained on Concrete Compressive Strength.

(2015) and the image **NYU Depth v2** dataset (Nathan Silberman & Fergus, 2012) where the goal is to predict the image depth per pixel. All inputs are rescaled with zero mean and unit variance. We also scale the output target for regression. Further details are given in the appendix.

**Metrics.** Beyond the target prediction error, we evaluate model uncertainty estimation using calibration and OOD detection scores. Furthermore, we report the inference speed. Further results including histograms with uncertainty estimates or latent space visualization are presented in appendix. *Target error:* We use the accuracy **(Accuracy)** for classification and the Root Mean Squared Error **(RMSE)** for regression and count prediction. *Calibration:* Calibration scores aim at evaluating if the error from the predicted target distribution corresponds to the empirical error achieved by the model (e.g. is a class prediction with a predicted confidence $p_c = 80\%$ correct $80\%$ of the time?). For classification, we use the Brier score **(Brier)** (Gneiting & Raftery, 2007). For regression and count prediction, we use the absolute difference between the percentile $p$ and the percentage $p_{\text{pred}}$ of target lying in the confidence interval $I_p = [0, \frac{p}{2}] \cup [1 - \frac{p}{2}, 1]$ under the predicted target distribution. We compute a single calibration score by summing the square difference for $p \in \{0.1, \dots, 0.9\}$ i.e. $\sqrt{\sum_p (p - p_{\text{pred}})^2}$ **(Calibration)** (Kuleshov et al., 2018). *OOD detection:* We evaluate how the uncertainty scores enable to detect OOD data using the area under the precision-recall curve (AUC-PR). We show further results using the area under the receiver operating characteristic curve (AUC-ROC) in appendix. We use two different uncertainty measures: the negative entropy of the predicted target distribution accounting for the aleatoric uncertainty **(Alea. OOD)** and the predicted evidence **(Epist. OOD)**. Similarly to (Charpentier et al., 2020; Malinin & Gales, 2019; Ovadia et al., 2019), we use four different types of OOD samples: Unseen datasets (KMNIST Clanuwat et al. (2018), Fashion-MNIST (Xiao et al., 2017), SVHN (Netzer et al., 2011), LSUN (Yu et al., 2015), CelebA (Liu et al., 2015), KITTI (Geiger et al., 2013)), left-out data (classes 9/10 for Sensorless Drive, winter/spring/autumn seasons for Bike Sharing), out-of-domain data not normalized in $[0, 1]$ (**OODom**) and dataset shifts (corrupted CIFAR-10 (Hendrycks & Dietterich, 2019)).

### 4.2 Results

**Classification.** We show results for the tabular dataset Sensorless Drive with unbounded input domain in Tab. 2, and the image datasets MNIST, FMNIST and CIFAR-10 with bounded input domain in Tab. 3 and appendix. Overall, for classification NatPN performs on par with best single-pass baselines (i.e. $12/30$ top-1 scores, $25/30$ top-2 scores) and NatPE performs the best among multiple-pass models (i.e. $28/30$ top-1 scores). A single NatPN achieves accuracy and calibration performance on par with the most calibrated baselines, namely PostNet and R-PriorNet which requires one

| | RMSE | Calibration | LSUN Alea. | LSUN Epist. | KITTI Alea. | KITTI Epist. | OODom Alea. | OODom Epist. |
|---|---|---|---|---|---|---|---|---|
| **Dropout** | 46.95 | 4.03 | *95.29 / 97.74 | 83.89 / 83.22 | 98.07 | 84.90 | 74.40 | *100.00 |
| **EvReg** | ***28.88** | ***1.05** | 58.70 / 56.71 | 70.19 / 64.02 | 56.60 | 62.67 | 75.43 | 56.39 |
| **NatPN** | 29.72 | 1.14 | **94.13 / *98.67** | ***89.08 / *90.56** | ***98.93** | ***93.15** | ***100.00** | ***100.00** |

Table 6: Regression results on NYU Depth v2 with Normal target distribution. RMSE is in cm. OOD scores on LSUN are reported on the held-out classes 'classrooms' (left) and 'churches' (right).

normalizing flow per class or training OOD data. Further, NatPE consistently improves accuracy and calibration performances of a single NatPN which underlines the benefit of aggregating multiple models predictions for accuracy and calibration (Lakshminarayanan et al., 2017). Without requiring OOD data during training, both NatPN and NatPE achieve excellent OOD detection scores w.r.t. all OOD types. This strongly suggests that NatPN does *not* suffer from the flaws in Choi et al. (2019); Nalisnick et al. (2019; 2020). In particular, NatPN and NatPE achieve almost perfect OODom scores contrary to all other baselines except PostNet. This observation aligns well with the theoretical guarantee of NatPN far from training data (see Thm. 1) which also applies to each NatPE member. The similar performance of NatPN and PostNet for classification is intuitively explained by their akin design: both models perform density estimation on a low-dimensional latent space. Similarly to Charpentier et al. (2020), we compute the average confidence avg-conf $= \frac{1}{N}\sum_i^N n^{(i)} = \frac{1}{N}\sum_i^N \alpha_0^{(i)}$ and then compare the average confidence change. The average confidence change is computed by taking the ratio of the average confidence of $N$ corrupted data at severity $s$ and the average confidence of $N$ clean data (i.e. the corrupted data at severity 0) i.e. $\frac{\text{avg-conf}_s}{\text{avg-conf}_0}$ for $s \in [\![1, 5]\!]$. NatPN maintains a competitive accuracy (Fig. 3, left) while assigning higher epistemic uncertainty as desired (Fig. 3, right). Baselines provide a slower relative confidence decay.

**Regression & Count Prediction.** We show the results for the Bike Sharing, the tabular UCI datasets and the image NYU Depth v2 datasets in Tab. 4, 5, 6. For the large NYU dataset, we compare against all baselines which require only a single model to be trained. Overall, NatPN outperforms other single-pass models for $23/26$ scores for regression, thus significantly improving calibration and OOD detection scores. Further, NatPN shows a strong improvement for calibration and OOD detection for count prediction with Poisson distributions among all models. Interestingly, all the models are less calibrated on the Bike Sharing dataset using a target Poisson distribution rather than a target Normal distribution. This suggests a mismatch of the Poisson distribution for this particular task. The almost perfect OODom scores of NatPN validate again Thm. 1 which also holds for regression.

**Inference Speed.** We show the average inference time per batch for all models on CIFAR-10 for classification and the NYU Depth v2 dataset for regression in Tab. 4. NatPN shows a significant improvement over Dropout and Ensemble which are both approximately five times slower since they require five forward passes for prediction. Notably, the NatPN speedup does not deteriorate target error and uncertainty scores. NatPN is slightly slower than R-PriorNet, EnD$^2$ and EvReg as they do not evaluate an additional normalizing flow. However, NatPN – which uses a single normalizing flow – is faster than PostNet – which scales linearly w.r.t. the number of classes since it evaluates one normalizing flow per class. Lastly, NatPN is the only single-pass model that can be used for *both* tasks.

| | CIFAR-10 (batch size 4,096) | NYU Depth v2 (batch size 4) |
|---|---|---|
| **Dropout** | $407.91 \pm 5.65$ | $650.96 \pm 0.22$ |
| **Ensemble** | $361.61 \pm 5.41$ | $649.78 \pm 0.18$ |
| **R-PriorNet** | $61.83 \pm 2.57$ | – |
| **EnD$^2$** | $61.83 \pm 2.57$ | – |
| **PostNet** | $88.56 \pm 0.06$ | – |
| **EvReg** | – | $129.88 \pm 0.75$ |
| **NatPN** | $75.64 \pm 0.04$ | $137.13 \pm 0.18$ |

Figure 4: Batched Inference Time (in ms), NVIDIA GTX 1080 Ti

## 5 Conclusion

We introduce a new family of models called Natural Posterior Network capable of efficient uncertainty estimation for the wide variety of tasks where the target distribution belongs to the exponential family (incl. classification, regression and count prediction). NatPN relies on the Bayes formula and the general form of exponential family distributions to perform a closed-form input-dependent posterior update over the target distribution. Further, an ensemble of NatPNs has a principled Bayesian combination interpretation. Theoretically, NatPN guarantees high uncertainty far from training data. Experimentally, NatPN achieves fast, versatile and high-quality uncertainty predictions with strong performance in calibration and OOD detection.

## 6 Ethics Statement

Accurate uncertainty estimation aims at improving trust in safety-critical domains subject to automation, and in a maintenance context where the underlying data distribution might slowly shift over time. In this regard, NatPN significantly improves the applicability of uncertainty estimation across a wide range of prediction domains (e.g. classification, regression, count prediction, etc) while maintaining a fast inference time. This could be particularly beneficial in industrial applications with time pressure and potential critical consequences (e.g. finance, medicine, policy decision making, etc).

Nonetheless, while NatPN achieves high-quality uncertainty estimation, there is always a risk that NatPN does not fully capture the real-world complexity e.g. for OOD data close to ID data. Furthermore, we raise awareness about two other risks of excessive trust related to the *Dunning-Kruger effect* (Kruger & Dunning, 2000): human excessive trust in Machine Learning model capacity, and human excessive trust in its own interpretation capacity. Therefore, we encourage practitioners to proactively confront the model design and its uncertainty estimates to desired behaviors in real-world use cases.

## 7 Reproducibility Statement

We provide all datasets at the following anonymous link [1]. In App. F, we give a detailed description for each dataset used in this paper. This description includes the task description, the dataset size, the input/output dimensions, the input/output preprossessing and the train/validation/test splits used in the experiments. We provide the model code at the following anonymous link [2]. In App. G, we give a detailed description for the architecture and grid search performed for each model used in this paper. Specifically, we provide an hyper-parameter study for NatPN in App. I.5, In App. H, we give a detailed description of the metrics used in the experiments. Finally, we provide a complete proof of Thm. 1, a detailed description of the Bayesian loss and relevant formulae for exponential family distributions in App. A, App. B and App. C.

## References

Alexander Amini, Wilko Schwarting, Ava Soleimany, and Daniela Rus. Deep evidential regression. *NeurIPS*, 2020.

Raman Arora, Amitabh Basu, Poorya Mianjy, and Anirbit Mukherjee. Understanding deep neural networks with rectified linear units. *ICLR*, 2018.

Marin Biloš, Bertrand Charpentier, and Stephan Günnemann. Uncertainty on asynchronous time event prediction. *NeurIPS*, 2019.

Christopher M Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

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.

Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight uncertainty in neural networks. *ICML*, 2015.

L. D. Brown. *Fundamentals of Statistical Exponential Families: With Applications in Statistical Decision Theory*. Institute of Mathematical Statistics, 1986.

David M Chan, Roshan Rao, Forrest Huang, and John F Canny. Gpu accelerated t-distributed stochastic neighbor embedding. *JDPC*, 2019.

Bertrand Charpentier, Daniel Zügner, and Stephan Günnemann. Posterior network: Uncertainty estimation without ood samples via density-based pseudo-counts. *NeurIPS*, 2020.

[1] `https://natural-posterior-network.s3.eu-central-1.amazonaws.com/data.zip`

[2] `https://natural-posterior-network.s3.eu-central-1.amazonaws.com/code.tar.gz`

Hugh Chipman, Edward George, and Robert Mcculloch. Bayesian ensemble learning. *NeurIPS*, 2007.

Hyunsun Choi, Eric Jang, and Alexander A Alemi. Generative ensembles for robust anomaly detection. *ICLR*, 2019.

Tarin Clanuwat, Mikel Bober-Irizar, Asanobu Kitamoto, Alex Lamb, Kazuaki Yamamoto, and David Ha. Deep learning for classical japanese literature. *arXiv preprint arXiv:1812.01718*, 2018.

Ylvisaker D. Diaconis P. Conjugate priors for exponential families. *The Annals of Statistics*, 1979.

J. Dix. Existence of the limit at infinity for a function that is integrable on the half line. *Ros-Hulman Undergraduate Mathematics Journal*, 2013.

Dheeru Dua and Casey Graff. UCI machine learning repository. 2017.

Michael W. Dusenberry, Ghassen Jerfel, Yeming Wen, Yi-An Ma, Jasper Snoek, Katherine Heller, Balaji Lakshminarayanan, and Dustin Tran. Efficient and scalable bayesian neural nets with rank-1 factors. *ICML*, 2020.

David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. *NeurIPS*, 2014.

Hadi Fanaee-T and João Gama. Event labeling combining ensemble detectors and background knowledge. *Progress in Artificial Intelligence*, 2014.

Sebastian Farquhar, Lewis Smith, and Yarin Gal. Liberty or depth: Deep bayesian neural nets do not need complex weight posterior approximations. *NeurIPS*, 2020.

Angelos Filos, Sebastian Farquhar, Aidan N. Gomez, Tim G. J. Rudner, Zachary Kenton, Lewis Smith, Milad Alizadeh, Arnoud de Kroon, and Yarin Gal. A systematic comparison of bayesian deep learning robustness in diabetic retinopathy tasks. *arXiv preprint arXiv:1912.10481*, 2019.

Andrew Y. K. Foong, David R. Burt, Yingzhen Li, and Richard E. Turner. On the expressiveness of approximate inference in bayesian neural networks. *NeurIPS*, 2020.

Yarin Gal. *Uncertainty in Deep Learning*. PhD thesis, University of Cambridge, 2016.

Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. *ICML*, 2016.

Jochen Gast and Stefan Roth. Lightweight probabilistic deep networks. *CVPR*, 2018.

Jakob Gawlikowski, Cedrique Rovile Njieutcheu Tassi, Mohsin Ali, Jongseok Lee, Matthias Humt, Jianxiang Feng, Anna Kruspe, Rudolph Triebel, Peter Jung, Ribana Roscher, Muhammad Shahzad, Wen Yang, Richard Bamler, and Xiao Xiang Zhu. A survey of uncertainty in deep neural networks. *arXiv preprint arXiv: 2107.03342*, 2021.

Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *IJRR*, 2013.

Andrew Gelman, John B. Carlin, Hal S. Stern, and Donald B. Rubin. *Bayesian Data Analysis*. Chapman and Hall/CRC, 2004.

Mathieu Germain, Karol Gregor, Iain Murray, and Hugo Larochelle. Made: Masked autoencoder for distribution estimation. *ICML*, 2015.

Tilmann Gneiting and Adrian E Raftery. Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association*, 2007.

Alex Graves. Practical variational inference for neural networks. *NeurIPS*, 2011.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks. *ICML*, 2017.

Matthias Hein, Maksym Andriushchenko, and Julian Bitterwolf. Why relu networks yield high-confidence predictions far away from the training data and how to mitigate the problem. *CVPR*, 2019.

Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *ICLR*, 2019.

Jose Miguel Hernandez-Lobato and Ryan Adams. Probabilistic backpropagation for scalable learning of bayesian neural networks. *ICML*, 2015.

Pavel Izmailov, Sharad Vikram, Matthew D Hoffman, and Andrew Gordon Gordon Wilson. What are bayesian neural network posteriors really like? *ICML*, 2021.

Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision? *NeurIPS*, 2017.

Hyun-Chul Kim and Zoubin Ghahramani. Bayesian classifier combination. *AISTATS*, 2012.

Polina Kirichenko, Pavel Izmailov, and Andrew Gordon Wilson. Why normalizing flows fail to detect out-of-distribution data. *NeurIPS*, 2020.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

Justin Kruger and David Dunning. Unskilled and unaware of it: How difficulties in recognizing one's own incompetence lead to inflated self-assessments. *JPSP*, 2000.

Volodymyr Kuleshov, Nathan Fenner, and Stefano Ermon. Accurate uncertainties for deep learning using calibrated regression. *ICML*, 2018.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. *NeurIPS*, 2017.

Balaji Lakshminarayanan, Dustin Tran, Jeremiah Liu, Shreyas Padhy, Tania Bedrax-Weiss, and Zi Lin. Simple and principled uncertainty estimation with deterministic deep learning via distance awareness. *NeurIPS*, 2020.

Charline Le Lan and Laurent Dinh. Perfect density models cannot guarantee anomaly detection. *arXiv preprint arXiv:2012.03808*, 2020.

Yann LeCun, Corinna Cortes, and CJ Burges. Mnist handwritten digit database. 2010.

Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. *ICCV*, 2015.

Wesley J Maddox, Pavel Izmailov, Timur Garipov, Dmitry P Vetrov, and Andrew Gordon Wilson. A simple baseline for bayesian uncertainty in deep learning. *NeurIPS*, 2019.

Andrey Malinin and Mark Gales. Predictive uncertainty estimation via prior networks. *NeurIPS*, 2018.

Andrey Malinin and Mark Gales. Reverse kl-divergence training of prior networks: Improved uncertainty and adversarial robustness. *NeurIPS*, 2019.

Andrey Malinin, Sergey Chervontsev, Ivan Provilkov, and Mark Gales. Regression prior networks. *arXiv preprint arXiv:2006.11590*, 2020a.

Andrey Malinin, Bruno Mlodozeniec, and Mark Gales. Ensemble distribution distillation. *ICLR*, 2020b.

Andrey Malinin, Neil Band, Ganshin, Alexander, German Chesnokov, Yarin Gal, Mark J. F. Gales, Alexey Noskov, Andrey Ploskonosov, Liudmila Prokhorenkova, Ivan Provilkov, Vatsal Raina, Vyas Raina, Roginskiy, Denis, Mariya Shmatova, Panos Tigas, and Boris Yangel. Shifts: A dataset of real distributional shift across multiple large-scale tasks. *arXiv preprint arXiv:2107.07455*, 2021.

Alexander Meinke and Matthias Hein. Towards neural networks that provably know when they don't know. *ICLR*, 2020.

K. Monteith, J. L. Carroll, K. Seppi, and T. Martinez. Turning bayesian model averaging into bayesian model combination. *IJCNN*, 2011.

Jooyoung Moon, Jihyo Kim, Younghak Shin, and Sangheum Hwang. Confidence-aware learning for deep neural networks. *ICML*, 2020.

Warren R. Morningstar, Cusuh Ham, Andrew G. Gallagher, Balaji Lakshminarayanan, Alexander A. Alemi, and Joshua V. Dillon. Density of states estimation for out-of-distribution detection. *arXiv preprint arXiv:2006.09273*, 2020.

Eric Nalisnick, Akihiro Matsukawa, Yee Whye Teh, Dilan Gorur, and Balaji Lakshminarayanan. Do deep generative models know what they don't know? *ICLR*, 2019.

Eric Nalisnick, Akihiro Matsukawa, Yee Whye Teh, and Balaji Lakshminarayanan. Detecting out-of-distribution inputs to deep generative models using typicality. *arXiv preprint arXiv:1906.02994*, 2020.

Jay Nandy, Wynne Hsu, and Mong-Li Lee. Towards maximizing the representation gap between in-domain & out-of-distribution examples. *NeurIPS*, 2020.

Pushmeet Kohli Nathan Silberman, Derek Hoiem and Rob Fergus. Indoor segmentation and support inference from rgbd images. *ECCV*, 2012.

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

Constantin P. Niculescu and Florin Popovici. A note on the behavior of integrable functions at infinity. *Journal of Mathematical Analysis and Applications*, 2011.

Frank Nielsen and Richard Nock. Entropies and cross-entropies of exponential families. *ICIP*, 2010.

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.

Yaniv Ovadia, Emily Fertig, Jie Ren, Zachary Nado, David Sculley, Sebastian Nowozin, Joshua Dillon, Balaji Lakshminarayanan, and Jasper Snoek. Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. *NeurIPS*, 2019.

George Papamakarios, Theo Pavlakou, and Iain Murray. Masked autoregressive flow for density estimation. *NeurIPS*, 2017.

Janis Postels, Francesco Ferroni, Huseyin Coskun, Nassir Navab, and Federico Tombari. Sampling-free epistemic uncertainty estimation using approximated variance propagation. *arXiv preprint arXiv:1908.00598*, 2019.

Amir Rahimi, Amirreza Shaban, Ching-An Cheng, Richard Hartley, and Byron Boots. Intra order-preserving functions for calibration of multi-class neural networks. *Advances in Neural Information Processing Systems*, 2020.

Rajesh Ranganath, Linpeng Tang, Laurent Charlin, and David Blei. Deep Exponential Families. *AISTATS*, 2015.

Danilo Jimenez Rezende and Shakir Mohamed. Variational inference with normalizing flows. *ICML*, 2015.

Hippolyt Ritter, Aleksandar Botev, and David Barber. A scalable laplace approximation for neural networks. *ICLR*, 2018.

Otto Rudolf Rocktäschel. *Methoden zur Berechnung der Gammafunktion für Komplexes Argument*. PhD thesis, Dresden University of Technology, 1922.

R. D. Rosenkrantz. *Prior Probabilities (1968)*. Springer Netherlands, 1989.

Murat Sensoy, Lance Kaplan, Federico Cerutti, and Maryam Saleki. Uncertainty-aware deep classifiers using generative models. *AAAI*, 2020.

John Shawe-Taylor and Robert C. Williamson. A pac analysis of a bayesian estimator. *COLT*, 1997.

Alexander Shekhovtsov and Boris Flach. Feed-forward propagation in probabilistic neural networks with categorical and max layers. *ICLR*, 2019.

Weishi Shi, Xujiang Zhao, Feng Chen, and Qi Yu. Multifaceted uncertainty estimation for label-efficient deep learning. *NeurIPS*, 2020.

Edwin Simpson, Stephen Roberts, Ioannis Psorakis, and Arfon Smith. Dynamic bayesian combination of multiple imperfect classifiers. *arXiv preprint arXiv:1206.1831*, 2012.

Hao Song, Tom Diethe, Meelis Kull, and Peter Flach. Distribution calibration for regression. *arXiv preprint arXiv:1905.06023*, 2019.

Joost van Amersfoort, Lewis Smith, Yee Whye Teh, and Yarin Gal. Uncertainty estimation using a single deep deterministic neural network. *ICML*, 2020.

Joost van Amersfoort, Lewis Smith, Andrew Jesson, Oscar Key, and Yarin Gal. On feature collapse and deep kernel learning for single forward pass uncertainty. *arXiv preprint arXiv:2102.11409*, 2021.

Hao Wang, Xingjian SHI, and Dit-Yan Yeung. Natural-parameter networks: A class of probabilistic neural networks. *NeurIPS*, 2016.

Yeming Wen, Dustin Tran, and Jimmy Ba. Batchensemble: an alternative approach to efficient ensemble and lifelong learning. *ICLR*, 2020.

Florian Wenzel, Jasper Snoek, Dustin Tran, and Rodolphe Jenatton. Hyperparameter ensembles for robustness and uncertainty quantification. *NeurIPS*, 2020.

E.T. Whittaker and G.N. Watson. *A Course of Modern Analysis*. Cambridge University Press, 1927.

Jim Winkens, Rudy Bunel, Abhijit Guha Roy, Robert Stanforth, Vivek Natarajan, Joseph R. Ledsam, Patricia MacWilliams, Pushmeet Kohli, Alan Karthikesalingam, Simon Kohl, Taylan Cemgil, S. M. Ali Eslami, and Olaf Ronneberger. Contrastive training for improved out-of-distribution detection. *arXiv preprint arXiv:2007.05566*, 2020.

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

Rikiya Yamashita, Mizuho Nishio, Richard Do, and Kaori Togashi. Convolutional neural networks: an overview and application in radiology. *Insights into Imaging*, 2018.

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

Arnold Zellner. Optimal information processing and bayes's theorem. *The American Statistician*, 1988.

Cheng Zhang, Judith Bütepage, Hedvig Kjellström, and Stephan Mandt. Advances in variational inference. *TPAMI*, 2018.

Shengjia Zhao, Tengyu Ma, and Stefano Ermon. Individual calibration with randomized forecasting. *ICML*, 2020a.

Xujiang Zhao, Feng Chen, Shu Hu, and Jin-Hee Cho. Uncertainty aware semi-supervised learning on graph data. *NeurIPS*, 2020b.