IMPROVING DIRICHLET PRIOR NETWORK FOR OUT-OF-DISTRIBUTION EXAMPLE DETECTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Determining the source of uncertainties in the predictions of AI systems are important. It allows the users to act in an informative manner to improve the safety of such systems, applied to the real-world sensitive applications. Predictive uncertainties can originate from the uncertainty in model parameters, data uncertainty or due to distributional mismatch between training and test examples. While recently, significant progress has been made to improve the predictive uncertainty estimation of deep learning models, most of these approaches either conflate the distributional uncertainty with model uncertainty or data uncertainty. In contrast, the Dirichlet Prior Network (DPN) can model distributional uncertainty distinctly by parameterizing a prior Dirichlet over the predictive categorical distributions. However, their complex loss function by explicitly incorporating KL divergence between Dirichlet distributions often makes the error surface ill-suited to optimize for challenging datasets with multiple classes. In this paper, we present an improved DPN framework by proposing a novel loss function using the standard cross-entropy loss along with a regularization term to control the sharpness of the output Dirichlet distributions from the network. Our proposed loss function aims to improve the training efficiency of the DPN framework for challenging classification tasks with large number of classes. In our experiments using synthetic and real datasets, we demonstrate that our DPN models can distinguish the distributional uncertainty from other uncertainty types. Our proposed approach significantly improves DPN frameworks and outperform the existing OOD detectors on CIFAR-10 and CIFAR-100 dataset while also being able to recognize distributional uncertainty distinctly.

1 INTRODUCTION

Deep neural networks (DNNs) have achieved impeccable success to address various real world tasks (Simonyan & Zisserman, 2014a; Hinton et al., 2012; Litjens et al., 2017). However, despite impressive, and ever-improving performance in various supervised learning tasks, DNNs tend to make over-confident predictions for every input. Predictive uncertainties of DNNs can be confronted from three different factors such as *model uncertainty, data uncertainty* and *distributional uncertainty* (Malinin & Gales, 2018). *Model or epistemic uncertainty* captures the uncertainty in estimating the model parameters, conditioning on training data (Gal, 2016). This uncertainty can be explained away given enough training data. *Data or aleatoric uncertainty* is originated from the inherent complexities of the training data, such as class overlap, label noise, homoscedastic and heteroscedastic noise (Gal, 2016). *Distributional uncertainty or dataset shift* arises due to the distributional mismatch between the training and test examples (Quionero-Candela et al., 2009; Malinin & Gales, 2018). In this case, the network is unfamiliar with the test data and hence should not confidently make predictions. The ability to separately model these three types of predictive uncertainty is important, as it enables the users to take appropriate actions depending on the source of uncertainty. For example, in the active learning scenario, distributional uncertainty indicates that the classifier requires additional data for training. On the other hand, for various real-world applications where the cost of an error is high, such as in autonomous vehicle control, medical, financial and legal fields, the source of uncertainty informs whether an input requires manual intervention.

Recently notable progress has been made to detect OOD images. Bayesian neural network based approaches conflate the distributional uncertainty through model uncertainty (Hernandez-Lobato &

Adams, 2015; Gal, 2016). However, since obtaining the true posterior distribution for the model parameters are intractable, the success of these approaches depends on the chosen prior distribution over parameters and the nature of approximations. Here, the predictive uncertainties can be measured by using an an ensemble of multiple stochastic forward passes using dropouts from a single DNN (Monti-Carlo Dropout or MCDP) (Gal & Ghahramani, 2016) or by ensembling results from multiple DNNs (Lakshminarayanan et al., 2017) and computing their mean and spread. On the other hand, most of the non-Bayesian approaches model their distributional uncertainty with data uncertainty. These approaches can explicitly train the network in a multi-task fashion incorporating both in-domain and OOD examples to produce sharp and flat predictive posteriors respectively (Lee et al., 2018a; Hendrycks et al., 2019). However, none of these approaches can robustly determine the source of uncertainty. Malinin & Gales (2018) introduced Dirichlet Prior Network (DPN) to distinctly model the distributional uncertainty from the other uncertainty types. A DPN classifier aims to produce sharp distributions to indicate low-order uncertainty for the in-domain examples and flat distributions for the OOD examples. However, their complex loss function, using the Kullback-Leibler (KL) divergence between Dirichlet distributions, results in the error surface to become poorly suited for optimization and makes it difficult efficiently train the DNN classifiers for challenging datasets with a large number of classes (Malinin & Gales, 2019).

In this work, we aim to improve the training efficiency of the DPN framework by proposing a novel loss function that also allows the distributional uncertainty to be modeled distinctly from both data uncertainty and model uncertainty. Instead of explicitly using Dirichlet distributions in the loss function, we propose to apply the standard cross-entropy loss on the softmax outputs along with a novel regularization term for the logit (pre-softmax activation) outputs. The proposed loss function can be also viewed from the perspective of the non-Bayesian frameworks (Lee et al., 2018a; Hendrycks et al., 2019) where the proposed regularizer presents an additional term to control the sharpness of the output Dirichlet distributions. In our experiments, we demonstrate that our proposed regularization term can effectively control the sharpness of the output Dirichlet distributions from the DPN to detect distributional uncertainties along with making the network scalable for more challenging datasets. We also demonstrate that the performance of our OOD detection model improves by training with noisy with OOD data. Our experimental results on CIFAR-10 and CIFAR-100 suggest that our proposed approach significantly improves the performance of the DPN framework for OOD detection and out-performs the recently proposed OOD detection techniques.

2 Related works

In Bayesian frameworks, the predictive uncertainty of a classification model, trained on a finite dataset, $\mathcal{D}_{in} = \{\boldsymbol{x}_i, y_i\}_{i=1}^{N} \sim P_{in}(\boldsymbol{x}, y)$, is expressed in terms of data (aleatoric) and model (epistemic) uncertainty (Gal, 2016). For an input $\boldsymbol{x}^*$, the predictive uncertainty is expressed as:

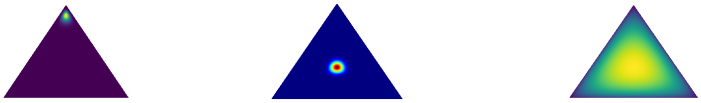
$$p(\omega_c|\boldsymbol{x}^*, \mathcal{D}_{in}) = \int p(\omega_c|\boldsymbol{x}^*, \theta)\ p(\boldsymbol{\theta}|\mathcal{D}_{in})\ d\boldsymbol{\theta} \tag{1}$$

Here, $\boldsymbol{x}$ and y represents the images and the corresponding class labels, sampled from an underlying probability distribution $p_{in}(\boldsymbol{x}, y)$. Here, the data uncertainty, $p(\omega_c|\boldsymbol{x}^*, \theta)$ is described by the posterior distribution over class labels given model parameters, θ and model uncertainty, $p(\boldsymbol{\theta}|\mathcal{D}_{in})$, is given by the posterior distribution over parameters given the data, $\mathcal{D}_{in}$.

The expected distribution for predictive uncertainty, $p(\omega_c|\boldsymbol{x}^*, \mathcal{D}_{in})$ is obtained by marginalizing out $\boldsymbol{\theta}$. However, true posterior for $p(\boldsymbol{\theta}|\mathcal{D}_{in})$ is intractable. Approaches such as Monte-Carlo dropout (MCDP) (Gal & Ghahramani, 2016), Langevin Dynamics (Welling & Teh, 2011), explicit ensembling (Lakshminarayanan et al., 2017) approximate the integral in eq. 1 as:

$$p(\omega_c|\boldsymbol{x}^*, \mathcal{D}_{in}) \approx \frac{1}{M} \sum_{m=1}^{M} p(\omega_c|\boldsymbol{x}^*, \boldsymbol{\theta}^{(m)}) \qquad \boldsymbol{\theta}^{(m)} \sim q(\boldsymbol{\theta}) \tag{2}$$

where, $\boldsymbol{\theta}^{(m)}$ is sampled from an explicit or implicit variational approximation, $q(\boldsymbol{\theta})$ of the true posterior $p(\boldsymbol{\theta}|\mathcal{D}_{in})$. Each $p(\omega_c|\boldsymbol{x}^*, \theta^{(m)})$ represents a categorical distribution, $\boldsymbol{\mu} = [\mu_1, \cdots, \mu_k] = [p(y = \omega_1), \cdots, p(y = \omega_K)]$ over the class labels and the ensemble can be visualized as a collection of points on the simplex. While for a confident prediction, the ensemble is expected to sharply appear in one corner of the simplex, the flatly spread ensembles cannot determine whether the uncertainty is due to data or distributional uncertainty. Furthermore, for standard DNNs, with millions



(a) Confident prediction (b) Data uncertainty (c) Distributional uncertainty

Figure 1: Desired behavior of a DPN to indicate the three different uncertainties.

of parameters, it becomes even harder to find an appropriate prior distribution and inference scheme to estimate the posterior distribution of the model. Dirichlet Prior Network (DPN) is introduced to explicitely model the distributional uncertainty by parameterizing a Dirichlet distribution over a simplex (Malinin & Gales, 2018). More discussions about DPN is presented in section 3.1.

Alternatively, non-Bayesian frameworks derive their measure of uncertainties using the predictive posteriors obtained from DNNs. Lee et al. (2018a) and Hendrycks et al. (2019) introduce new components in their loss functions to explicitly incorporate OOD data for training. DeVries & Taylor (2018) append an auxiliary branch onto a pre-trained classifier to derive the OOD score. Shalev et al. (2018) use multiple semantic dense representations as the target label to train the OOD detection network. Several recent works such as (Lee et al., 2018b; Liang et al., 2018) have demonstrated that by tweaking the input images during inference using adversarial perturbations can enhance the performance of a DNN for OOD detection (Goodfellow et al., 2014b). However, their discriminative scores are achieved by tailoring the parameters for each OOD distributions during test time, which is not possible for real-world OOD examples. Hein et al. (2019) propose an adversarial training (Madry et al., 2018) like approach to produce lower confident predictions for OOD examples. However, while these models can identify the total predictive uncertainties, they can not robustly determine whether the source of uncertainty is due to an in-domain input in a region of class overlap or an OOD example far away from the training distribution.

3 Proposed Methodology

This section first describes the DPN framework and the difficulties of the existing modeling techniques to scale DPNs for challenging datasets. We then present our improved version DPN by proposing a novel loss function to address these difficulties while allowing to model the distributional uncertainty distinctly from the model and data uncertainty.

3.1 Dirichlet Prior Network

A DPN for classification directly parametrizes a prior Dirichlet distribution over the categorical output distributions on a simplex (Malinin & Gales, 2018). For in-domain examples, a DPN attempts to produce sharp Dirichlet in one corner of the simplex, when it is confident in its predictions (Figure 1a). It should produce a sharp distribution in the middle of the simplex to indicate the data (low-order) uncertainty for the in-domain example with a high degree of noise or belongs to a class overlapping region (Figure 1b). In contrast, for OOD examples, a DPN should produce a flat Dirichlet over the simplex to indicate the distributional (high-order) uncertainty for the input (Figure 1c). Here, the data uncertainty is expressed by the point-estimate categorical distribution $\boldsymbol{\mu}$ while the distributional uncertainty is described using the distribution over the predictive categorical i.e $p(\boldsymbol{\mu}|\boldsymbol{x}^*, \theta)$. The overall predictive uncertainty is expressed as:

$$p(\omega_c|\boldsymbol{x}^*, \mathcal{D}) = \int\int p(\omega_c|\boldsymbol{\mu})\ p(\boldsymbol{\mu}|\boldsymbol{x}^*, \boldsymbol{\theta})\ p(\boldsymbol{\theta}|\mathcal{D})\ d\boldsymbol{\mu}\ d\boldsymbol{\theta} \tag{3}$$

This expression forms a three layered hierarchy of uncertainties: a large model uncertainty, $p(\theta|\mathcal{D})$ would induce a large variation in distributional uncertainty in $p(\boldsymbol{\mu}|\boldsymbol{x}^*, \theta)$ and a large degree of uncertainty for $\boldsymbol{\mu}$ leads to higher data uncertainty. DPN framework is consistent with the existing approaches, where an additional layer of uncertainty is included to capture the distributional uncertainty. For example, marginalization of $\boldsymbol{\mu}$ in Eqn. 3 will reproduce Eqn. 1 while loose the control over the sharpness of the output Dirichlet distributions. The marginalization of $\boldsymbol{\theta}$ produces the expected estimation of data and distributional uncertainty given model uncertainty as:

$$p(\omega_c|\boldsymbol{x}^*, \mathcal{D}) = \int p(\omega_c|\boldsymbol{\mu}) \Big[\int p(\boldsymbol{\mu}|\boldsymbol{x}^*, \boldsymbol{\theta})\ p(\boldsymbol{\theta}|\mathcal{D}) d\boldsymbol{\theta} \Big] d\boldsymbol{\mu} = \int p(\omega_c|\boldsymbol{\mu})\ p(\boldsymbol{\mu}|\boldsymbol{x}^*, \mathcal{D}) d\boldsymbol{\mu} \tag{4}$$

However, Similar to eq. 1 marginalizing $\boldsymbol{\theta}$ is eq. 4 is also intractable. Since the model uncertainty is reducible given large training data, for simplicity, here we assume a dirac delta estimation for $\boldsymbol{\theta}$:

$$p(\boldsymbol{\theta}|\mathcal{D}) = \delta(\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}) \implies p(\boldsymbol{\mu}|\boldsymbol{x}^*, \mathcal{D}) \approx p(\boldsymbol{\mu}|\boldsymbol{x}^*, \boldsymbol{\theta}) \tag{5}$$

Constructing a DPN. A DPN constructs a Dirichlet distribution as a prior over the categorical distributions, which is parameterized by the concentration parameters, $\boldsymbol{\alpha} = \alpha_1, \cdots, \alpha_K$.

$$Dir(\boldsymbol{\mu}|\boldsymbol{\alpha}) = \frac{\Gamma(\alpha_0)}{\prod_{c=1}^{K} \Gamma(\alpha_c)} \prod_{c=1}^{K} \mu_c^{\alpha_c - 1}, \qquad \alpha_c > 0, \quad \alpha_0 = \sum_{c=0}^{K} \alpha_c \tag{6}$$

where, α_0 is called the *precision* of the Dirichlet. A larger value of α_0 produces sharper distributions to indicate low order uncertainties (fig 1a and 1b). A DPN, $f_{\hat{\theta}}$ produces the concentration parameters, $\boldsymbol{\alpha}$ and the posterior over class labels, $p(\omega_c|\boldsymbol{x}^*; \hat{\boldsymbol{\theta}})$, is given by the mean of the Dirichlet.

$$\boldsymbol{\alpha} = f_{\hat{\theta}}(\boldsymbol{x}^*) \qquad p(\boldsymbol{\mu}|x^*; \hat{\theta}) = Dir(\boldsymbol{\mu}|\boldsymbol{\alpha}) \qquad p(\omega_c|\boldsymbol{x}^*; \hat{\boldsymbol{\theta}}) = \int p(\omega_c|\boldsymbol{\mu}) \; p(\boldsymbol{\mu}|x^*; \hat{\boldsymbol{\theta}}) \; d\boldsymbol{\mu} \; = \frac{\alpha_c}{\alpha_0} \tag{7}$$

A standard DNN with the softmax activation function can be represented as a DPN where the concentration parameters are $\alpha_c = e^{z_c(\boldsymbol{x}^*)}$; $z_c(\boldsymbol{x}^*)$ is the pre-softmax (logit) output corresponding to the class, c for an input $\boldsymbol{x}^*$. The expected posterior probability of class label ω_c is given as:

$$p(\omega_c|\boldsymbol{x}^*; \hat{\boldsymbol{\theta}}) \; = \frac{\alpha_c}{\alpha_0} = \frac{e^{z_c(\boldsymbol{x}^*)}}{\sum_{c=1}^{K} e^{z_c(\boldsymbol{x}^*)}} \tag{8}$$

However, the mean of the Dirichlet is now *insensitive* to any arbitrary scaling of α_c. Hence, the precision of the Dirichlet, α_0, degrades under the standard cross-entropy loss. Malinin & Gales (2018) instead introduced a new loss function that explicitly minimizes the KL divergence between the output Dirichlet and a target Dirichlet to produce a predefined target precision value for the output Dirichlet distributions. For in-domain examples, the target distribution is chosen to be a sharp Dirichlet, $Dir(\boldsymbol{\mu}|\hat{\boldsymbol{\alpha}}_y)$, focusing on their ground truth classes while for OOD examples, a flat Dirichlet, $Dir(\boldsymbol{\mu}|\tilde{\boldsymbol{\alpha}})$ is selected.

$$\mathcal{L}(\theta) = \mathbb{E}_{P_{in}} KL[Dir(\boldsymbol{\mu}|\hat{\boldsymbol{\alpha}}_y)||p(\boldsymbol{\mu}|\boldsymbol{x}, \boldsymbol{\theta})] + \mathbb{E}_{P_{out}} KL[Dir(\boldsymbol{\mu}|\tilde{\boldsymbol{\alpha}})||p(\boldsymbol{\mu}|\boldsymbol{x}, \boldsymbol{\theta})] \tag{9}$$

where, P_{in} and P_{out} are the underlying distribution of in-domain and OOD training examples respectively. However, learning the model using sparse 1-hot continuous distributions for class labels, which are effectively a delta function, is challenging due to their complex loss function (eq. 9). Here, the error surface becomes poorly suited for optimization using the back-propagation algorithm (Malinin & Gales, 2019). Malinin & Gales (2018) tackle this problem by using label smoothing (Szegedy et al., 2016) or teacher-student training (Hinton et al., 2015a) to redistribute a small amount of probability density to each corner of the Dirichlet. This technique is found to work well for datasets with a fewer number of class labels. However, for more challenging datasets with a large number of classes, even these techniques cannot efficiently redistribute the probability densities at each corner and results in the target distribution to tend to a delta function. Hence, it becomes difficult to train the DPN to achieve competitive performances. Malinin & Gales (2019) have recently proposed to reverse the terms within the KL divergence in eq. 9 to improve the training efficiency of DPN models. This approach still requires to explicitly constrain the precision of the output Dirichlet distributions using an appropriately chosen hyper-parameter for training.

3.2 Improved Dirichlet Prior Network

We now propose an improved technique to model DPN by proposing a novel loss function using the standard cross-entropy loss along with a regularization term to control the precision of the output Dirichlet distribution from the network. As we have seen in equation 8, the precision of a Dirichlet distribution produced by the standard DNN is given as $\sum_{c=1}^{K} \exp z_c(\boldsymbol{x}^*)$. Hence, we can control the sharpness of the distribution by designing a regularization term that increases the sum of logit (pre-softmax) outputs for the in-domain examples to produce sharp distributions to indicate their lower uncertainties. For the OOD examples, the regularization term aims to decrease the sum of logit (pre-softmax) outputs to produce flat distributions to indicate distributional (higher-order) uncertainties. Hence, instead of explicitly constraining the precision the output Dirichlet with a specific hyper-parameter, we allow the network to appropriately produce the precision values for different input.

In this paper, we propose the regularization term as $\frac{1}{K}\sum_{c=1}^{K}\text{sigmoid}(z_c(\boldsymbol{x}))$ for controlling the sharpness of the output Dirichlet distribution. For in-domain examples, the loss function is given as:

$$\mathcal{L}_{in}(\boldsymbol{\theta}) = \mathbb{E}_{P_{in}}\Big[-\log p(y|\boldsymbol{x},\boldsymbol{\theta}) - \frac{\lambda_{in}}{K}\sum_{c=1}^{K}\text{sigmoid}(z_c(\boldsymbol{x}))\Big], \qquad \lambda_{in} > 0 \tag{10}$$

The constrain $\lambda_{in} > 0$ enforces the network to produce larger precision, α_0 and hence generate a sharper Dirichlet in one corner of the simplex. For OOD examples, the loss function is given as:

$$\mathcal{L}_{out}(\boldsymbol{\theta}) = \mathbb{E}_{P_{out}}\Big[\mathcal{H}_c(\mathcal{U}; p(\boldsymbol{\omega}|\boldsymbol{x},\boldsymbol{\theta})) - \frac{\lambda_{out}}{K}\sum_{c=1}^{K}\text{sigmoid}(z_c(\boldsymbol{x}))\Big], \qquad \lambda_{out} < \lambda_{in} \tag{11}$$

The constrain $\lambda_{out} < \lambda_{in}$ enforces the network to produce Dirichlet distributions with lower precisions for OOD examples compared to the in-domain examples. $\mathcal{U}$ denotes the uniform distribution over the class labels for OOD examples and $\mathcal{H}_c$ is the cross-entropy. We train the network in a multi-task fashion, where the overall loss function is given as:

$$\min_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}) = \mathcal{L}_{in}(\boldsymbol{\theta}) + \lambda\mathcal{L}_{out}(\boldsymbol{\theta}), \qquad \lambda > 0 \tag{12}$$

where, λ_{in} , λ_{out} and λ in Eq. 10, Eq. 11, and Eq. 12 represent user-defined hyper-parameters. The proposed loss function in Eq. 12 is also very closely related to non-Bayesian approaches, where by choosing λ_{in}, λ_{out} to zero we re-obtain similar loss functions as proposed by Lee et al. (2018a); Hendrycks et al. (2019). However, by setting λ_{in}, λ_{out} to zero, we lose control over the precision of the Dirichlet distribution that distinguishes distributional uncertainty from data uncertainty.

Our multi-task loss function in eq. 12 requires training samples from the in-domain distribution, P_{in} as well as from OOD P_{out}. However, since P_{out} is unknown, Lee et al. (2018a) propose to synthetically generate the OOD training samples from the boundary of in-domain region, P_{in} using generative models such as GAN (Goodfellow et al., 2014a). Alternatively, a different, easily available, real datasets can be used as OOD training examples. In practice, the latter approach is to found to be more effective for training the OOD detectors and has been applied for our experiments on vision datasets (Hendrycks et al., 2019).

4 Experimental Study

In this section, we experimentally demonstrate the importance of the DPN framework and its effectiveness using the proposed loss function using two sets of experiments. Our first experiment demonstrates the effectiveness of a DPN model using the proposed loss function using a synthetic dataset. Our second experiment on CIFAR10 and CIFAR100 presents a comparative study of our proposed method with the existing approaches and demonstrates the advantages compared to the original DPN framework(Malinin & Gales, 2018).

4.1 Synthetic Dataset

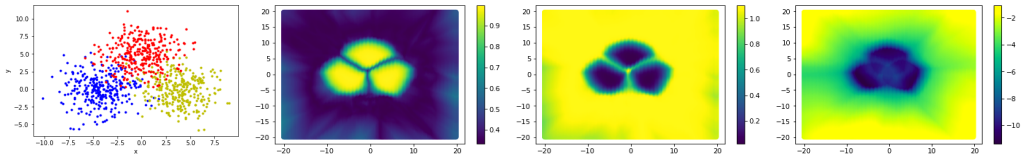
To demonstrate the effectiveness of our DPN framework using the proposed loss function, we design a simple dataset with three classes sampled from three different isotropic Gaussian distributions as shown in Figure 2(a). We select an isotropic co-variances, $\sigma^2 I$ with $\sigma = 4$ to ensure that the classes are overlapping. We train a small DPN with 2 hidden layers of 50 nodes each for the synthetic dataset. The hyper-parameters of our loss function, $\lambda_{in}, \lambda_{out}$ and λ are set to 1.0, 0.33 and 1.0.

Figure 2(b) and 2(c) represent the *total uncertainty measures* for each data point. These measures are derived from the expected predictive categorical distribution, $p(\omega_c|\boldsymbol{x}^*, D_{in})$ i.e by obtaining a complete marginalization of $\boldsymbol{\mu}$ and $\boldsymbol{\theta}$ in eq. 3. Here, the predictive categoricals are obtained as the point estimation from the network similar to a non-Bayesian framework (eq. 5). An uncertainty measure can be computed as the probability of the predicted class or *max probability* (Figure 2(b)):

$$max\mathcal{P} = \max_{c} p(\omega_c|\boldsymbol{x}^*, D_{in}) \tag{13}$$

Entropy of the predicted distribution, $\mathcal{H}[p(\omega_c|\boldsymbol{x}^*, D_{in})]$ can be also applied as a total uncertainty measure that produces low scores when the model is confident in its prediction (Figure 2(c)):

$$\mathcal{H}[p(\omega_c|\boldsymbol{x}^*, D_{in})] = -\sum_{c=1}^{K} p(\omega_c|\boldsymbol{x}^*, D_{in}) \ln p(\omega_c|\boldsymbol{x}^*, D_{in}) \tag{14}$$



(a) Synthetic Data (b) Max Probability (c) Entropy (d) Differential Entropy

Figure 2: Visualizing the uncertainties under different measures.

Max probability and *entropy* are the most frequently used uncertainty measures by the existing OOD detection model. However, since the predicted distribution is obtained by marginalizing $\boldsymbol{\mu}$ (eq. 3), these measures cannot capture the sharpness of the output Dirichlet for a DPN. As we can see in Figure 2(b) and 2(c), the overlapped in-domain data points and OOD points remain indistinguishable under max probability and entropy measures. This observation also indicates the limitation of the existing approaches and the uncertainty measures to differentiate between data and distributional uncertainties (Malinin & Gales, 2018). A DPN framework can overcome this limitation by using the *differential entropy* as an uncertainty measure that produces high scores for flat Dirichlet distributions (eq. 15).

$$\mathcal{H}[p(\boldsymbol{\mu}|\boldsymbol{x}^*, D_{in})] = -\int_{S^{K-1}} p(\boldsymbol{\mu}|\boldsymbol{x}^*, D_{in}) \ln p(\boldsymbol{\mu}|\boldsymbol{x}^*, D_{in}) \tag{15}$$

Figure 2(d) demonstrates that differential entropy can distinguish between in-domain and OOD examples. It also ensures that the network learned using our proposed loss function works perfectly as a DPN to produce sharp distributions for all in-domain examples and flat distributions for OOD examples. Additional details and results are provided in Appendix B.

4.2 Experiments on CIFAR-10 and CIFAR-100

We demonstrate the performance of DPN using our proposed loss function for OOD detection on CIFAR-10 and CIFAR-100 (Krizhevsky, 2009). These two datasets contain 32×32 natural colored images of $50,000$ training and $10,000$ testing examples For CIFAR-10, the images belong to 10 different classes while CIFAR-100 is a more challenging dataset, containing 100 image classes.

Evaluation of OOD detection methods. To evaluate the ability to detect OOD examples of our proposed models, we treat the OOD examples as the positive class and in-domain examples as a negative class and measure the OOD detection performance using two metrics: *area under the receiver operating characteristic curve (AUROC)* and *area under the precision-recall curve (AUPR)* (Hendrycks & Gimpel, 2016). The AUROC can be interpreted as the probability of an OOD example to produce a higher detection score than an in-domain example (Davis & Goadrich, 2006). Hence, a higher AUROC is desirable, and an uninformative detector produces an AUROC $\approx 50\%$. The AUPR is more informative when the positive class and negative class have greatly differing base rates, as it can take these base rates into account (Manning & Schütze, 1999).

Comparative Results and Analysis. In Table 1, we present a smaller set of comparative results to analyze our proposed approach using an in-domain and an OOD datasets. *Gaussian* is an artificially generated in-domain dataset where the test images of CIFAR-10 and CIFAR-100 are modified using Gaussian noises sampled from isotropic Gaussian distributions, $\mathcal{N}(0, \sigma^2 I)$ with $\sigma = 0.25$. *TinyImageNet (TIM)* is a real world image dataset (Li et al., 2017). In Appendix A, we present an expanded version of this comparative table for a wide range of OOD examples. In Appendix C, we provide the details of the OOD datasets and the comparative models.

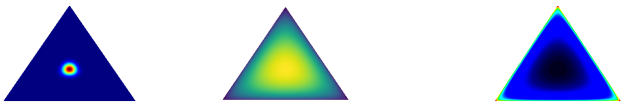
Since we have explicitly constrained on the logit outputs to produce smaller values for OOD examples, it is meaningful to define the *sum of the exponential of logits*, $\sum_{c=1}^{K} e^{z_c(\boldsymbol{x}^*)}$, as a new measure of predictive uncertainty for our proposed framework. Experimentally, we find that it often produces a better estimation for predictive uncertainties compared to the existing measures for our models.

Training Details. We train multiple DPN models using the proposed loss function, denoted as DPN$_{soft}$ for our analysis. For CIFAR-10, we train a VGG-16 model using CIFAR-10 training images as in-domain and CIFAR-100 training images as OOD examples (Simonyan & Zisserman,

2014b). For CIFAR-100, we use CIFAR-100 training images as in-domain and CIFAR-10 training images as OOD examples and train a DenseNet model with depth $= 55$, growth rate $= 12$ (Huang et al., 2017). We apply the standard data augmentation techniques such as rotation and translation for training. See Appendix C for additional details of our training.

Choosing the Hyper-parameters. Unlike Liang et al. (2018); Lee et al. (2018a) and similar to Malinin & Gales (2018); Hendrycks et al. (2019), we do not tune the hyper-parameters at testing phase for different D_{out}^{test}. Hence, the OOD examples remain unknown, as in a real-world scenario. For our experiments, we always set $\lambda = 0.5$ (eq. 12) similar to Hendrycks et al. (2019).

The hyper-parameters λ_{in} and λ_{out} controls the sharpness of the output Dirichlet from a DPN (see eq. 10 and 11). Since we want sharper distributions for in-domain data points and flatter distribution for OOD data points, we select $\lambda_{in} > 0$ and $\lambda_{in} > \lambda_{out}$. However, choosing $\lambda_{out} < 0$ would enforce the network to produce fractional values for α_c's (i.e $\alpha_c \in (0, 1)$) for OOD examples. This cause in the densities of the Dirichlet to be distributed in the edges of the simplex and produces a sharp distribution across the edges as shown in Figure 3(b). Hence, even though it indicates a higher-order distributional uncertainty, the differential entropy (eq. 15) can not capture this uncertainty and produce a low score, similar to an in-domain example.



(a) $\boldsymbol{\alpha} = (50, 50, 50)$ (b) $\boldsymbol{\alpha} = (1.5, 1.5, 1.5)$ (c) $\boldsymbol{\alpha} = (0.99, 0.99, 0.99)$

Figure 3: Dirichlet distributions become flatter as the precision is reduced (from (a) to (b)) and the densities move to the edges as the concentration parameters become fractional i.e $\alpha_c \in (0, 1)$.

For our experiments, we choose both positive and negative values for λ_{out} in our experiments. Here, λ_{in} and λ_{out} are chosen as $\lambda_{in} = (1 - \beta)$ and $\lambda_{out} = (\frac{1}{\#class} - \beta)$. We train two different sets of DPN models using $\beta = 0.5$ (i.e negative λ_{out}) and 0.0 (i.e positive λ_{out}).

As we can see in Table 1, for TinyImageNet, *differential entropy* (D. Ent) scores are found to be uninformative for our DPN$_{soft}(\beta = 0.5)$. While the total uncertainty measures are consistently producing high AUROC and AUPR scores, differential entropy has produced low scores which are often less than 50% for AUROC. These scores support the assertion that for OOD examples, DPN$_{soft}(\beta = 0.5)$ is producing sharp distribution along the edge of the simplex to indicate high distributional uncertainty. However, differential entropy measure failed to capture this uncertainty as it cannot distinguish between the sharp Dirichlet distributions in the middle and along the edges of the simplex. On the other hand, for $\beta = 0.0$, the differential entropy produces similar AUROC and AUPR scores as total uncertainty measures and successfully captures the distributional uncertainty.

In contrast, for Gaussian dataset, our DPN$_{soft}(\beta = 0.0)$ models achieve high AUROC and AUPR scores under the total uncertainty measures while producing lower score under differential entropy. These scores indicate that the predictive uncertainty in the *Gaussian dataset* is due to the *data uncertainty rather than distributional uncertainty*. This is also expected because, here, we have generated the dataset by applying noises in the in-domain test examples. Notably, existing non-DPN models cannot detect the cause of this uncertainty.

Fine-tuning with noisy OOD images. After training the networks using clean images, we fine-tune our DPN models using noisy OOD images for a few epochs in the end. Here, the idea is to add small noise without distorting the original OOD training images to further expose the network into the out of distribution space.

Table 1 also demonstrate the performance after fine-tuning our DPN$_{soft}$ models with noisy OOD training images where the noises are samples from isotropic Gaussian distributions $\mathcal{N}(0, \sigma^2 I)$ with three different isotropic variances: $\sigma = 0.0$ (i.e no noise), $\sigma = 0.01$ and $\sigma = 0.05$. As we can see that the performance of DPN$_{soft}(\sigma = 0.01)$ is improved from DPN$_{soft}(\sigma = 0.0)$; however it often declined for DPN$_{soft}(\sigma = 0.05)$.

Comparison with existing models. In Table 1, we compare the performance our approach with standard DNN as baseline (Hendrycks & Gimpel (2016)), MCDP (Gal & Ghahramani, 2016),

Table 1: Comparative results of OOD example detection for CIFAR-10 and CIFAR-100. Expanded version of this table along with a wide range of OOD datasets are provided in Appendix A.

$\mathcal{D}_{in}$	$\mathcal{D}_{out}^{test}$	Methods	AUROC				AUPR			
			Max.P	Ent.	$\sum e^{z_c(\boldsymbol{x}^*)}$	D. Ent	Max.P	Ent.	$\sum e^{z_c(\boldsymbol{x}^*)}$	D. Ent
CIFAR-10	Gaussian	Baseline	79.4	79.6	-	-	65.4	65.7	-	-
		MCDP	79.2	79.5	-	-	64.9	65.2	-	-
		ODIN	89.9	-	-	-	82.6	-	-	-
		OE	96.82	97.0	-	-	92.7	93.1	-	-
		DPN$_{soft}(\beta: 0.0, \sigma: 0.0)$	99.8	99.9	99.4	66.1	99.4	99.6	97.7	53.9
		DPN$_{soft}(\beta: 0.0, \sigma: 0.01)$	99.8	99.9	99.7	66.5	99.1	99.5	98.6	55.1
		DPN$_{soft}(\beta: 0.0, \sigma: 0.05)$	**100**	**100**	**100**	64.2	**100**	**100**	99.9	54.3
		DPN$_{soft}(\beta: 0.5, \sigma: 0.0)$	99.6	99.6	97.8	4.7	99.1	99.1	93.1	31.5
		DPN$_{soft}(\beta: 0.5, \sigma: 0.01)$	99.7	99.8	96.8	3.5	98.7	99.4	89.1	31.4
		DPN$_{soft}(\beta: 0.5, \sigma: 0.05)$	**100**	**100**	99.7	2.6	**100**	**100**	98.6	30.74
CIFAR-10	TIM	Baseline	88.8	89.4	-	-	85.1	86.7	-	-
		MCDP	88.5	89.2	-	-	84.7	86.1	-	-
		ODIN	94.4	-	-	-	93.8	-	-	-
		OE	98.0	98.0	-	-	97.9	97.9	-	-
		DPN$_{Dir}$	94.3	94.3		94.6	94.0	94.0		94.2
		DPN$_{soft}(\beta: 0.0, \sigma: 0.0)$	97.6	97.7	97.6	97.6	97.5	97.6	97.5	97.5
		DPN$_{soft}(\beta: 0.0, \sigma: 0.01)$	98.5	98.5	98.4	98.5	98.4	98.5	98.3	98.4
		DPN$_{soft}(\beta: 0.0, \sigma: 0.05)$	97.1	97.4	97.6	97.7	97.2	97.5	97.7	97.8
		DPN$_{soft}(\beta: 0.5, \sigma: 0.0)$	98.7	98.8	96.7	6.8	98.6	98.7	92.8	32.5
		DPN$_{soft}(\beta: 0.5, \sigma: 0.01)$	99.0	**99.1**	96.3	6.8	98.5	**98.9**	90.7	32.2
		DPN$_{soft}(\beta: 0.5, \sigma: 0.05)$	97.9	98.2	98.2	30.9	97.9	98.2	98.1	54.1
CIFAR-100	Gaussian	Baseline	75.2	75.1	-	-	66.8	64.9	-	-
		MCDP	77.5	75.8	-	-	67.5	63.4	-	-
		ODIN	58.3	-	-	-	51.0	-	-	-
		OE	91.3	93.0	-	-	82.4	83.1	-	-
		DPN$_{soft}(\beta: 0.0, \sigma: 0.0)$	78.8	81.4	90.6	57.0	67.3	69.8	80.3	48.3
		DPN$_{soft}(\beta: 0.0, \sigma: 0.01)$	96.0	96.9	99.2	68.2	94.5	94.9	98.7	54.9
		DPN$_{soft}(\beta: 0.0, \sigma: 0.05)$	99.9	99.9	**100**	63.8	99.8	99.6	**100**	52.0
		DPN$_{soft}(\beta: 0.5, \sigma: 0.0)$	92.4	92.4	95.1	43.9	83.8	82.5	87.1	42.4
		DPN$_{soft}(\beta: 0.5, \sigma: 0.01)$	96.3	96.3	99.0	22.7	91.7	90.7	97.7	35.4
		DPN$_{soft}(\beta: 0.5, \sigma: 0.05)$	98.5	99.4	**100**	20.1	97.6	98.5	**100**	35.1
CIFAR-100	TIM	Baseline	74.9	76.3	-	-	71.1	73.1	-	-
		MCDP	78.9	81.0	-	-	75.4	78.0	-	-
		ODIN	83.8	-	-	-	81.4	-	-	-
		OE	86.5	88.0	-	-	82.8	83.0	-	-
		DPN$_{soft}(\beta: 0.0, \sigma: 0.0)$	89.9	90.3	91.1	90.7	86.1	86.4	85.4	83.6
		DPN$_{soft}(\beta: 0.0, \sigma: 0.01)$	96.5	97.4	**98.8**	98.0	97.1	97.8	**98.9**	95.2
		DPN$_{soft}(\beta: 0.0, \sigma: 0.05)$	95.8	96.7	98.0	96.7	96.4	97.2	98.2	92.9
		DPN$_{soft}(\beta: 0.5, \sigma: 0.0)$	85.6	87.7	91.2	81.8	82.6	83.8	84.7	82.3
		DPN$_{soft}(\beta: 0.5, \sigma: 0.01)$	92.6	94.1	97.0	49.5	92.8	93.9	96.1	63.0
		DPN$_{soft}(\beta: 0.5, \sigma: 0.05)$	95.7	96.8	98.7	43.5	96.3	97.3	98.8	60.7

DPN$_{Dir}$ (Malinin & Gales, 2018), ODIN (Liang et al., 2018) and OE (Hendrycks et al., 2019). We use the same architecture as our DPN$_{soft}$ models for all the competitive models. OE models are trained using the set of in-domain and OOD training images with their proposed loss function (Hendrycks et al., 2019). Note that, since non-DPN methods do not explicitly model the logit outputs, the differential entropy or $\sum_{c=1}^{K} e^{z_c(\boldsymbol{x}^*)}$ measures are not meaningful to define for these models (Malinin & Gales, 2018). For DPN$_{Dir}$, we compare with the results reported by Malinin & Gales (2018) for CIFAR-10, while the framework failed to train for CIFAR-100 with 100 classes. Due to the unavailability of codes and results (under the same settings), we could not compare our model with Malinin & Gales (2019). As we can see in Table 1, our DPN$_{soft}$ models significantly improved the performance of the DPN framework and consistently out-performed the existing OOD detection models along with able to distinguish distributional uncertainty from other uncertainty types.

5 Conclusion

In this paper, we present an improved DPN framework to improve the training efficiency for complex classification tasks with large number of classes. We propose a novel loss function using standard cross-entropy loss along with a regularization term for that allows controls the sharpness of the Dirichlet distributions. In our experiments using synthetic and real datasets, we demonstrate that our proposed framework works perfectly as a DPN to distinguish the distributional uncertainty from other uncertainty types. We demonstrate that the OOD detection performance of our DPN models can be improved by training with noisy OOD examples. Our proposed approach significantly improves DPN frameworks and outperform the existing OOD detectors on CIFAR-10 and CIFAR-100 dataset while also being able to recognize distributional uncertainty distinctly.

REFERENCES

M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, , and A. Vedaldi. Describing textures in the wild. In *Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2014.

Jesse Davis and Mark Goadrich. The relationship between precision-recall and roc curves. In *Proceedings of the 23rd International Conference on Machine Learning*, ICML '06, 2006.

Terrance DeVries and Graham W Taylor. Learning confidence for out-of-distribution detection in neural networks. *arXiv preprint arXiv:1802.04865*, 2018.

Yarin Gal. *Uncertainty in deep learning*. PhD thesis, PhD thesis, University of Cambridge, 2016.

Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *Proceedings of The 33rd International Conference on Machine Learning (ICML)*, Proceedings of Machine Learning Research, 2016.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pp. 2672–2680, 2014a.

Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014b.

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. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 41–50, 2019.

Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. *arXiv preprint arXiv:1610.02136*, 2016.

Dan Hendrycks, Mantas Mazeika, and Thomas Dietterich. Deep anomaly detection with outlier exposure. In *International Conference on Learning Representations*, 2019. URL `https://openreview.net/forum?id=HyxCxhRcY7`.

Jose Miguel Hernandez-Lobato and Ryan Adams. Probabilistic backpropagation for scalable learning of bayesian neural networks. In *Proceedings of the 32nd International Conference on Machine Learning*, 2015.

Geoffrey Hinton, Li Deng, Dong Yu, George Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Brian Kingsbury, et al. Deep neural networks for acoustic modeling in speech recognition. *IEEE Signal processing magazine*, 29, 2012.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015a.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015b.

Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

Alex Krizhevsky. Learning multiple layers of features from tiny images. 2009. URL `http://www.cs.toronto.edu/~kriz/cifar.html`.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, 2017.

Kimin Lee, Honglak Lee, Kibok Lee, and Jinwoo Shin. Training confidence-calibrated classifiers for detecting out-of-distribution samples. In *International Conference on Learning Representations*, 2018a. URL `https://openreview.net/forum?id=ryiAv2xAZ`.

Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. In *Advances in Neural Information Processing Systems*, pp. 7167–7177, 2018b.

Fei-Fei Li, Andrej Karpathy, and Justin Johnson. Tiny imagenet visual recognition challenge. 2017. URL https://tiny-imagenet.herokuapp.com/.

Shiyu Liang, Yixuan Li, and R. Srikant. Enhancing the reliability of out-of-distribution image detection in neural networks. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=H1VGkIxRZ.

Geert Litjens, Thijs Kooi, Babak Ehteshami Bejnordi, Arnaud Arindra Adiyoso Setio, Francesco Ciompi, Mohsen Ghafoorian, Jeroen Awm Van Der Laak, Bram Van Ginneken, and Clara I Sánchez. A survey on deep learning in medical image analysis. *Medical image analysis*, 42: 60–88, 2017.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *ICLR*, 2018.

Andrey Malinin and Mark Gales. Predictive uncertainty estimation via prior networks. In *Proceedings of the 32Nd International Conference on Neural Information Processing Systems*, NIPS'18, pp. 7047–7058, USA, 2018. Curran Associates Inc. URL http://dl.acm.org/citation.cfm?id=3327757.3327808.

Andrey Malinin and Mark Gales. Reverse kl-divergence training of prior networks: Improved uncertainty and adversarial robustness. In *Neural Information Processing Systems*, 2019.

Christopher D. Manning and Hinrich Schütze. *Foundations of Statistical Natural Language Processing*. MIT Press, Cambridge, MA, USA, 1999. ISBN 0-262-13360-1.

Joaquin Quionero-Candela, Masashi Sugiyama, Anton Schwaighofer, and Neil D Lawrence. *Dataset shift in machine learning*. The MIT Press, 2009.

Gabi Shalev, Yossi Adi, and Joseph Keshet. Out-of-distribution detection using multiple semantic label representations. In *Advances in Neural Information Processing Systems*, pp. 7375–7385, 2018.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014a.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014b. URL http://arxiv.org/abs/1409.1556.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2818–2826, 2016.

Max Welling and Yee Whye Teh. Bayesian learning via stochastic gradient langevin dynamics. In *Proceedings of the 28th International Conference on International Conference on Machine Learning*, ICML'11, pp. 681–688, USA, 2011. Omnipress. ISBN 978-1-4503-0619-5. URL http://dl.acm.org/citation.cfm?id=3104482.3104568.

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.

Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

A EXPANDED RESULTS

Table 2: Expanded results of OOD image detection for CIFAR-10.

$\mathcal{D}_{out}^{test}$	Methods	AUROC				AUPR			
		Max.P	Ent.	$\sum e^{z_c(\boldsymbol{x}^*)}$	D. Ent	Max.P	Ent.	$\sum e^{z_c(\boldsymbol{x}^*)}$	D. Ent
Gaussian	Baseline	79.4	79.6	-	-	65.4	65.7	-	-
	MCDP	79.2	79.5	-	-	64.9	65.2	-	-
	ODIN	89.9	-	-	-	82.6	-	-	-
	OE	96.82	97.0	-	-	92.7	93.1	-	-
	DPN$_{soft}(\beta: 0.0, \sigma: 0.0)$	99.8	99.9	99.4	66.1	99.4	99.6	97.7	53.9
	DPN$_{soft}(\beta: 0.0, \sigma: 0.01)$	99.8	99.9	99.7	66.5	99.1	99.5	98.6	55.1
	DPN$_{soft}(\beta: 0.0, \sigma: 0.05)$	**100**	**100**	**100**	64.2	**100**	**100**	99.9	54.3
	DPN$_{soft}(\beta: 0.5, \sigma: 0.0)$	99.6	99.6	97.8	4.7	99.1	99.1	93.1	31.5
	DPN$_{soft}(\beta: 0.5, \sigma: 0.01)$	99.7	99.8	96.8	3.5	98.7	99.4	89.1	31.4
	DPN$_{soft}(\beta: 0.5, \sigma: 0.05)$	**100**	**100**	99.7	2.6	**100**	**100**	98.6	30.74
TIM	Baseline	88.8	89.4	-	-	85.1	86.7	-	-
	MCDP	88.5	89.2	-	-	84.7	86.1	-	-
	ODIN	94.4	-	-	-	93.8	-	-	-
	OE	98.0	98.0	-	-	97.9	97.9	-	-
	DPN$_{Dir}$	94.3	94.3		94.6	94.0	94.0		94.2
	DPN$_{soft}(\beta: 0.0, \sigma: 0.0)$	97.6	97.7	97.6	97.6	97.5	97.6	97.5	97.5
	DPN$_{soft}(\beta: 0.0, \sigma: 0.01)$	98.5	98.5	98.4	98.5	98.4	98.5	98.3	98.4
	DPN$_{soft}(\beta: 0.0, \sigma: 0.05)$	97.1	97.4	97.6	97.7	97.2	97.5	97.7	97.8
	DPN$_{soft}(\beta: 0.5, \sigma: 0.0)$	98.7	98.8	96.7	6.8	98.6	98.7	92.8	32.5
	DPN$_{soft}(\beta: 0.5, \sigma: 0.01)$	99.0	**99.1**	96.3	6.8	98.5	**98.9**	90.7	32.2
	DPN$_{soft}(\beta: 0.5, \sigma: 0.05)$	97.9	98.2	98.2	30.9	97.9	98.2	98.1	54.1
LSUN	Baseline	90.2	91.0	-	-	86.6	88.6	-	-
	MCDP	90.3	91.1	-	-	86.8	88.9	-	-
	ODIN	96.6	-	-	-	96.2	-	-	-
	OE	97.9	98.0	-	-	97.6	97.7	-	-
	DPN$_{Dir}$	94.4	94.4		94.6	93.3	93.4		93.3
	DPN$_{soft}(\beta: 0.0, \sigma: 0.0)$	97.7	97.8	97.7	97.7	97.3	97.4	97.3	97.3
	DPN$_{soft}(\beta: 0.0, \sigma: 0.01)$	98.7	98.7	98.6	98.6	98.5	98.5	98.3	98.3
	DPN$_{soft}(\beta: 0.0, \sigma: 0.05)$	97.3	97.7	97.9	98.0	97.2	97.6	97.7	97.8
	DPN$_{soft}(\beta: 0.5, \sigma: 0.0)$	98.8	98.8	97.1	6.2	98.6	98.6	93.8	32.4
	DPN$_{soft}(\beta: 0.5, \sigma: 0.01)$	**99.2**	**99.2**	96.7	5.5	98.7	**98.9**	91.8	31.8
	DPN$_{soft}(\beta: 0.5, \sigma: 0.05)$	97.9	98.2	98.4	33.0	97.7	98.1	98.3	56.0
Places365	Baseline	89.4	90.0	-	-	95.6	96.2	-	-
	MCDP	89.3	90.2	-	-	95.6	96.2	-	-
	ODIN	95.4	-	-	-	98.5	-	-	-
	OE	97.9	98.0	-	-	99.4	99.4	-	-
	DPN$_{soft}(\beta: 0.0, \sigma: 0.0)$	97.7	97.8	97.7	97.7	99.3	99.3	99.3	99.3
	DPN$_{soft}(\beta: 0.0, \sigma: 0.01)$	98.8	98.8	98.7	98.7	99.6	99.6	99.6	99.6
	DPN$_{soft}(\beta: 0.0, \sigma: 0.05)$	97.3	97.6	97.8	97.8	99.2	99.3	99.4	99.4
	DPN$_{soft}(\beta: 0.5, \sigma: 0.0)$	98.7	98.7	96.7	6.7	99.6	99.6	97.9	60.7
	DPN$_{soft}(\beta: 0.5, \sigma: 0.01)$	99.2	**99.3**	96.5	5.4	99.6	**99.7**	97.3	59.5
	DPN$_{soft}(\beta: 0.5, \sigma: 0.05)$	97.9	98.2	98.3	29.5	99.4	99.5	99.5	77.1
Textures	Baseline	88.6	89.0	-	-	74.9	77.0	-	-
	MCDP	87.6	88.0	-	-	73.7	75.8	-	-
	ODIN	94.9	-	-	-	90.5	-	-	-
	OE	99.2	99.7	-	-	98.6	98.5	-	-
	DPN$_{soft}(\beta: 0.0, \sigma: 0.0)$	99.5	99.5	99.5	99.5	98.8	99.0	99.0	99.0
	DPN$_{soft}(\beta: 0.0, \sigma: 0.01)$	99.5	99.5	99.3	99.4	98.9	98.9	98.6	98.7
	DPN$_{soft}(\beta: 0.0, \sigma: 0.05)$	99.4	99.5	99.5	**99.6**	99.0	99.1	99.2	**99.3**
	DPN$_{soft}(\beta: 0.5, \sigma: 0.0)$	99.5	99.5	96.7	4.6	99.1	99.1	85.6	21.3
	DPN$_{soft}(\beta: 0.5, \sigma: 0.01)$	99.4	99.5	96.5	5.5	98.0	98.8	84.3	21.4
	DPN$_{soft}(\beta: 0.5, \sigma: 0.05)$	99.1	99.3	98.7	24.7	98.3	98.7	97.3	38.3

B EXPERIMENTAL DETAILS ON SYNTHETIC DATASETS

B.1 EXPERIMENTAL SETUP

The three classes of our synthetic dataset is constructed by sampling from three different isotropic Gaussian distributions with means of $(-4, 0)$, $(4, 0)$ and $(0, 5)$ and isotropic variances of $\sigma = 4$. We

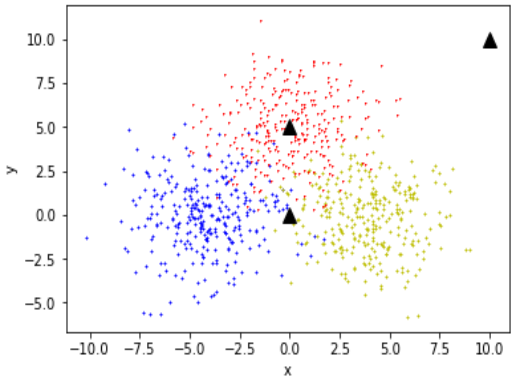
Table 3: Expanded results of OOD image detection for CIFAR-100.

$\mathcal{D}_{out}^{test}$	Methods	AUROC				AUPR			
		Max.P	Ent.	$\sum e^{z_c(\boldsymbol{x}^*)}$	D. Ent	Max.P	Ent.	$\sum e^{z_c(\boldsymbol{x}^*)}$	D. Ent
Gaussian	Baseline	75.2	75.1	-	-	66.8	64.9	-	-
	MCDP	77.5	75.8	-	-	67.5	63.4	-	-
	ODIN	58.3	-	-	-	51.0	-	-	-
	OE	91.3	93.0	-	-	82.4	83.1	-	-
	$\text{DPN}_{soft}(\beta: 0.0, \sigma: 0.0)$	78.8	81.4	90.6	57.0	67.3	69.8	80.3	48.3
	$\text{DPN}_{soft}(\beta: 0.0, \sigma: 0.01)$	96.0	96.9	99.2	68.2	94.5	94.9	98.7	54.9
	$\text{DPN}_{soft}(\beta: 0.0, \sigma: 0.05)$	99.9	99.9	**100**	63.8	99.8	99.6	**100**	52.0
	$\text{DPN}_{soft}(\beta: 0.5, \sigma: 0.0)$	92.4	92.4	95.1	43.9	83.8	82.5	87.1	42.4
	$\text{DPN}_{soft}(\beta: 0.5, \sigma: 0.01)$	96.3	96.3	99.0	22.7	91.7	90.7	97.7	35.4
	$\text{DPN}_{soft}(\beta: 0.5, \sigma: 0.05)$	98.5	99.4	**100**	20.1	97.6	98.5	**100**	35.1
TIM	Baseline	74.9	76.3	-	-	71.1	73.1	-	-
	MCDP	78.9	81.0	-	-	75.4	78.0	-	-
	ODIN	83.8	-	-	-	81.4	-	-	-
	OE	86.5	88.0	-	-	82.8	83.0	-	-
	$\text{DPN}_{soft}(\beta: 0.0, \sigma: 0.0)$	89.9	90.3	91.1	90.7	86.1	86.4	85.4	83.6
	$\text{DPN}_{soft}(\beta: 0.0, \sigma: 0.01)$	96.5	97.4	**98.8**	98.0	97.1	97.8	**98.9**	95.2
	$\text{DPN}_{soft}(\beta: 0.0, \sigma: 0.05)$	95.8	96.7	98.0	96.7	96.4	97.2	98.2	92.9
	$\text{DPN}_{soft}(\beta: 0.5, \sigma: 0.0)$	85.6	87.7	91.2	81.8	82.6	83.8	84.7	82.3
	$\text{DPN}_{soft}(\beta: 0.5, \sigma: 0.01)$	92.6	94.1	97.0	49.5	92.8	93.9	96.1	63.0
	$\text{DPN}_{soft}(\beta: 0.5, \sigma: 0.05)$	95.7	96.8	98.7	43.5	96.3	97.3	98.8	60.7
LSUN	Baseline	78.9	80.4	-	-	74.2	75.9	-	-
	MCDP	83.2	85.4	-	-	79.0	81.5	-	-
	ODIN	87.8	-	-	-	84.6	-	-	-
	OE	90.6	91.6	-	-	86.5	86.1	-	-
	$\text{DPN}_{soft}(\beta: 0.0, \sigma: 0.0)$	91.6	92.6	93.3	92.6	88.3	88.3	87.3	85.4
	$\text{DPN}_{soft}(\beta: 0.0, \sigma: 0.01)$	98.9	99.2	**99.7**	98.9	99.0	99.3	**99.7**	96.1
	$\text{DPN}_{soft}(\beta: 0.0, \sigma: 0.05)$	96.0	96.8	98.2	96.8	96.6	97.2	98.4	93.0
	$\text{DPN}_{soft}(\beta: 0.5, \sigma: 0.0)$	91.8	93.2	94.8	76.4	88.1	88.8	88.8	80.5
	$\text{DPN}_{soft}(\beta: 0.5, \sigma: 0.01)$	95.6	96.6	98.0	44.7	95.1	95.8	97.1	59.7
	$\text{DPN}_{soft}(\beta: 0.5, \sigma: 0.05)$	96.6	97.5	99.1	41.4	97.1	97.9	99.1	59.2
Places365	Baseline	76.6	78.1	-	-	90.2	91.0	-	-
	MCDP	80.9	83.1	-	-	92.2	93.3	-	-
	ODIN	86.4	-	-	-	94.7	-	-	-
	OE	88.8	90.1	-	-	95.1	95.0		-
	$\text{DPN}_{soft}(\beta: 0.0, \sigma: 0.0)$	90.5	91.3	92.6	92.0	96.0	96.0	95.9	94.9
	$\text{DPN}_{soft}(\beta: 0.0, \sigma: 0.01)$	97.8	98.4	**99.3**	98.5	99.4	99.6	**99.8**	98.6
	$\text{DPN}_{soft}(\beta: 0.0, \sigma: 0.05)$	95.9	96.7	98.1	96.7	98.9	99.1	99.5	97.6
	$\text{DPN}_{soft}(\beta: 0.5, \sigma: 0.0)$	89.5	91.2	93.7	77.4	95.6	95.9	96.0	92.8
	$\text{DPN}_{soft}(\beta: 0.5, \sigma: 0.01)$	94.7	95.9	97.8	43.4	98.3	98.6	99.1	81.7
	$\text{DPN}_{soft}(\beta: 0.5, \sigma: 0.05)$	96.3	97.2	98.9	39.6	99.0	99.2	99.7	80.7
Textures	Baseline	60.0	60.2	-	-	43.4	43.4	-	-
	MCDP	64.0	64.3	-	-	46.0	45.6	-	-
	ODIN	63.4	-	-	-	48.9	-	-	-
	OE	73.6	74.8	-	-	58.3	58.4	-	-
	$\text{DPN}_{soft}(\beta: 0.0, \sigma: 0.0)$	80.5	81.8	85.0	83.7	66.6	66.9	71.3	67.2
	$\text{DPN}_{soft}(\beta: 0.0, \sigma: 0.01)$	84.3	86	92.9	90.9	81.3	83.3	90.9	84.1
	$\text{DPN}_{soft}(\beta: 0.0, \sigma: 0.05)$	89.2	90.7	94.5	92.8	86.2	88.3	93.1	84.5
	$\text{DPN}_{soft}(\beta: 0.5, \sigma: 0.0)$	69.1	70.8	82.0	61.3	57.0	58.3	68.7	49.7
	$\text{DPN}_{soft}(\beta: 0.5, \sigma: 0.01)$	77.6	79.4	90.1	38.9	72.2	74.2	86.6	36.0
	$\text{DPN}_{soft}(\beta: 0.5, \sigma: 0.05)$	85.3	87.5	**96.4**	31.8	82.1	85.2	**95.7**	36.1

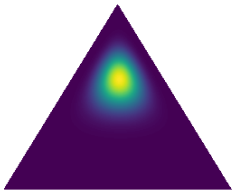
sample 200 training data points from each distributions for each classes. We also sample 600 OOD training examples from an uniform distribution of $\mathcal{U}([-15, 15], [-13, 17])$.

We train a neural network with 2 hidden layers with 50 nodes each and $relu$ activation function. The network is trained for $2,500$ epochs using stochastic gradient descent (SGD) optimization with constant learning rate of 0.01. The hyper-parameter values for $\lambda_{in}, \lambda_{out}$ and λ are set to 1.0, 0.33 and 1.0.

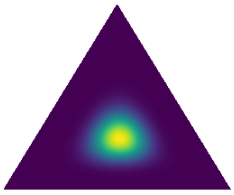
To demonstrate that our model produces sharp distributions for in-domain examples and flat distributions for OOD examples, we visualize the output Dirichlet distributions for three data points in Figure 4.



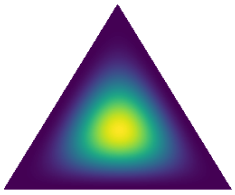
(a) Synthetic in-domain training data (dots of different colors to indicate different classes) and test inputs (shown in black triangles).



(b) Data point: $(0, 5)$
Entropy: 0.027; Diff. Ent.: -9.97



(c) Data point: $(0, 0)$
Entropy: 0.8; Diff. Ent.: -9.0



(d) Data point: $(10, 10)$
Entropy: 1.09; Diff. Ent.: -3.36

Figure 4: Visualizing the data point uncertainties under different measures on a synthetic dataset with 3 classes. Our DPN model aim to produce sharp Dirichlet distribution for all in-domain data points and hence differential entropy measure easily distinguishes them from the OOD examples.

As we can see in Figure 4(b), for $(0, 5)$, an in-domain examples near class 3, the network produces a sharp distribution near one corner of the simplex. The entropy and differential entropy measures are low for this data point to indicate a confident prediction.

Figure 4(c), demonstrates that as we choose $(0, 0)$, a sample from the overlapping region of 3 classes, the network produces a sharp distribution in the middle of the simplex. While the entropy is high, the differential entropy is low for this data points to indicate data uncertainty.

Finally, as we choose an OOD example at $(10, 10)$, the network produces a flat Dirichlet distribution as shown in Figure 4(d). Both entropy and differential entropy measures are high for this data point to indicate distributional uncertainty .

C Experimental Details on CIFAR-10 and CIFAR-100

C.1 Experimental Setup

For our experiments on CIFAR-10, we train a VGG-16 model with CIFAR-10 as the in-domain and CIFAR-100 as the OOD training data (Simonyan & Zisserman (2014b)). For CIFAR-100, we train a DenseNet with depth $= 55$, growth rate $= 12$ and CIFAR-100 as the in-domain and CIFAR-10 as the OOD training data (Huang et al. (2017)). We trained multiple DPN$_{soft}$ models using our proposed loss functions with different hyper-parameters.

For CIFAR-10, VGG-16 is trained for 250 epochs using stochastic gradient descent (SGD) optimization. We set the initial learning rate to 0.1 are reduced the learning rate by half after every 20 epochs. For CIFAR-100, Densenet(55, 12) is trained using the same setup as proposed by Huang et al. (2017).

After training the models with clean in-domain and OOD images, we further fine-tune the models using noisy OOD images for 50 epochs with learning rate of 0.0001. Here the noises are chosen

form an isotropic Gaussian distribution, $\mathcal{N}(0, \sigma^2 I)$. We have experimented with three different values of σ as $\{0.0, 0.01, 0.05\}$ to introduce different level of noises. The test accuracies achieved by our models on CIFAR-10 and CIFAR-100 datasets, are summarized in Table 4.

Table 4: Test accuracies on CIFAR-10 and CIFAR-100 datasets.

	CIFAR-10	CIFAR-100
Baseline (Standard DNN)	94.1	76.3
DPN$_{soft}(\beta: 0.0, \sigma: 0.0)$	94.0	75.7
DPN$_{soft}(\beta: 0.0, \sigma: 0.01)$	93.6	75.6
DPN$_{soft}(\beta: 0.0, \sigma: 0.05)$	93.7	75.7
DPN$_{soft}(\beta: 0.5, \sigma: 0.0)$	94.1	76.2
DPN$_{soft}(\beta: 0.5, \sigma: 0.01)$	93.8	76.3
DPN$_{soft}(\beta: 0.5, \sigma: 0.05)$	94.1	76.3

In Table 5, we present the performance for detecting the misclassified test examples of our proposed models, DPN_{soft}. As we can see, our models have achieved high AUROC and AUPR scores under total uncertainty measures such as maximum probability and entropy. On the other hand, the differential entropy has consistently produced low (uninformative) scores for distinguishing the misclassified examples for our proposed DPN framework. This indicates that our DPN models are always producing sharp Dirichlet distributions for the in-distribution examples. For the confident predictions, these models produce sharp Dirichlets in one corner of the simplex and hence achieves higher scores for entropy and lower scores for differential entropy. For the misclassified examples, they tend to produce sharp Dirichlets in the middle of the simplex and hence achieves lower scores for entropy as well as for differential entropy. 5, we can easily infer that here, the source of the predictive uncertainty for the in-domain dataset is due to the data uncertainty rather than distributional uncertainty.

Table 5: Misclassification Detection.

CIFAR-10 Dataset						
Methods	AUROC			AUPR		
	Max.P	Ent.	D. Ent	Max.P	Ent.	D. Ent
Baseline (Standard DNN)	93.2	93.3	-	43.0	46.6	-
DPN$_{soft}(\beta: 0.0, \sigma: 0.0)$	91.7	91.3	55.2	37.0	35.7	6.2
DPN$_{soft}(\beta: 0.0, \sigma: 0.01)$	91.3	90.8	54.7	35.0	33.3	6.4
DPN$_{soft}(\beta: 0.0, \sigma: 0.05)$	93.4	93.2	53.4	44.7	42.5	6.2
DPN$_{soft}(\beta: 0.5, \sigma: 0.0)$	92.0	91.6	32.3	36.8	34.8	4.6
DPN$_{soft}(\beta: 0.5, \sigma: 0.01)$	91.3	90.8	39.1	37.1	35.3	4.8
DPN$_{soft}(\beta: 0.5, \sigma: 0.05)$	93.0	92.8	49.4	39.8	38.3	5.4

CIFAR-100 Dataset						
Methods	AUROC			AUPR		
	Max.P	Ent.	D. Ent	Max.P	Ent.	D. Ent
Baseline (Standard DNN)	86.8	87.0	-	63.6	64.7	-
DPN$_{soft}(\beta: 0.0, \sigma: 0.0)$	86.5	85.4	50.2	60.7	57.1	21.8
DPN$_{soft}(\beta: 0.0, \sigma: 0.01)$	86.9	86.2	50.7	61.5	58.5	22.1
DPN$_{soft}(\beta: 0.0, \sigma: 0.05)$	87.3	86.7	47.7	64.6	61.7	21.0
DPN$_{soft}(\beta: 0.5, \sigma: 0.0)$	86.8	85.6	52.0	62.1	58.2	22.1
DPN$_{soft}(\beta: 0.5, \sigma: 0.01)$	86.7	85.7	52.7	61.1	57.5	22.3
DPN$_{soft}(\beta: 0.5, \sigma: 0.05)$	87.5	87.0	49.6	64.0	61.5	21.0

C.2 Out of Distribution Datasets

We use a wide range of OOD dataset to evaluate the performance of our proposed OOD detection models. The OOD images are resized to 32×32 before applying to the network. For our evaluations, we use the entire set of OOD images as described in the following.

1. **TinyImageNet (TIM)** (Li et al. (2017)). This is a subset of Imagenet dataset. It contains $10,000$ test images from 200 different image classes.
2. **LSUN** (Yu et al. (2015)). The Large-scale Scene UNderstanding dataset (LSUN) contains $10,000$ images of 10 different scene categories.
3. **Places 365** (Zhou et al. (2017)) consists of 36500 images of 365 scene categories.
4. **Textures** (Cimpoi et al. (2014)) contains 5640 textural images in the wild belonging to 47 categories.
5. **Gaussian Noise.** This is an artificially generated dataset obtained by modifying the in-domain test images using Gaussian noises sampled from isotropic Gaussian distribution, $\mathcal{N}(0, \sigma^2 I)$ with $\sigma = 0.25$.

C.3 Details of Competitive Systems

We compare the performance of our models with standard DNN as baseline model (Hendrycks & Gimpel (2016)), the Bayesian framework, monti-carlo dropout (MCDP) (Gal & Ghahramani (2016)), DPN$_D ir$ using the loss function proposed by Malinin & Gales (2018), non-Bayesian frameworks such as ODIN (Liang et al. (2018)) and outlier exposure (OE) by Hendrycks et al. (2019). We use the same architecture as DPN$_{softmax}$ for the competitive models. For DPN$_{Dirichlet}$, we could not reproduce the same performance as given in Malinin & Gales (2018) and hence use their reported results for CIFAR-10 for our comparison.

For MCDP, we use the standard DNN model with randomly dropping the nodes during test time. The predictive categorical distributions are obtained by averaging the outputs for 10 iterations.

ODIN applies the standard DNN models trained only using in-domain training examples for OOD detection. During testing phase, it perturbs the input images using FGSM adversarial attack (Goodfellow et al. (2014b))and softmax activation function by incorporating the temperature hyper-parameter (Hinton et al. (2015b)). Maximum Probability score is then applied for their uncertainty measure. They propose to use different hyper-parameters for different OOD examples. However in practice, the source of expected OOD examples cannot be known. Hence, for our comparisons, we always set the perturbation size to 0.002 and temperature to 1000.

OE models are trained using the proposed loss function by Hendrycks et al. (2019). Here, we use the same training setup as applied for our DPN$_{soft}$ models: CIFAR-10 classifiers are trained using CIFAR-10 training images as in-domain examples and CIFAR-100 training images as OOD examples. For CIFAR-100, the OE models are trained using CIFAR-10 training images as OOD examples.

D Differential Entropy Measure for Dirichlet Prior Network

Differential Entropy of a Dirichlet distribution can be calculated as follows (Malinin & Gales, 2018):

$$\begin{aligned}\mathcal{H}[p(\boldsymbol{\mu}|\boldsymbol{x}^*, D_{in})] &= -\int_{S^{K-1}} p(\boldsymbol{\mu}|\boldsymbol{x}^*, D_{in}) \ln p(\boldsymbol{\mu}|\boldsymbol{x}^*, D_{in}) \\ &= \sum_{c=1}^{K} \ln \Gamma(\alpha_c) - \ln \Gamma(\alpha_0) - \sum_{c=1}^{K} (\alpha_c - 1)(\psi(\alpha_c) - \psi(\alpha_0))\end{aligned} \tag{16}$$

Note that, α_c is a function of x^*. Γ and ψ denotes the Gamma and digamma functions respectively.