Towards Maximizing the Representation Gap between In-Domain & Out-of-Distribution Examples

Jay Nandy **Wynne Hsu** **Mong Li Lee**
National University of Singapore
{jaynandy,whsu,leeml}@comp.nus.edu.sg

Abstract

Among existing uncertainty estimation approaches, Dirichlet Prior Network (DPN) distinctly models different predictive uncertainty types. However, for in-domain examples with high data uncertainties among multiple classes, even a DPN model often produces indistinguishable representations from the out-of-distribution (OOD) examples, compromising their OOD detection performance. We address this shortcoming by proposing a novel loss function for DPN to maximize the *representation gap* between in-domain and OOD examples. Experimental results demonstrate that our proposed approach consistently improves OOD detection performance.

1 Introduction

Deep neural network (DNN) based models have achieved impeccable success to address various real-world tasks [1, 2, 3]. However, when these intelligent systems fail, they do not provide any explanation or warning. Predictive uncertainty estimation has emerged as an important research direction to inform users about possible wrong predictions and allow users to react in an informative manner, thus improving their reliability.

Predictive uncertainties of DNNs can come from three different sources: *model uncertainty, data uncertainty,* and *distributional uncertainty* [4, 5]. Model uncertainty or epistemic uncertainty captures the uncertainty in estimating the model parameters, conditioning on training data [4]. Data uncertainty (or aleatoric uncertainty) arises from the natural complexities of the underlying distribution, such as class overlap, label noise, homoscedastic and heteroscedastic noise, etc [4]. Distributional uncertainty or dataset shift arises due to the distributional mismatch between the training and test examples, that is, the test data is *out-of-distribution (OOD)* [6, 5].

It is useful to determine the sources of predictive uncertainties. In active learning, distributional uncertainty indicates that the classifier requires additional data for training. For real-world applications, where the cost of errors are high, such as in autonomous vehicles [7], medical diagnosis [8], financial, and legal fields [9], the source of uncertainty can allow manual intervention in an informed way.

Notable progress has been made for predictive uncertainty estimation. Bayesian neural network-based models conflate the distributional uncertainty through model uncertainty [10, 4, 11, 12, 13]. However, since the true posterior for their model parameters are intractable, their success depend on the nature of approximations. In contrast, non-Bayesian approaches can explicitly train the network in a multi-task fashion, incorporating both in-domain and OOD examples to produce sharp and uniform categorical predictions respectively [14, 15]. However, these approaches cannot robustly determine the source of predictive uncertainty [5]. In particular, the presence of high data uncertainty among multiple classes leads them to produce uniform categorical predictions for in-domain examples, often making them indistinguishable from the OOD examples.

Dirichlet Prior Network (DPN) separately models different uncertainty types by producing sharp uni-modal Dirichlet distributions for in-domain examples, and flat Dirichlet distributions for OOD

34th Conference on Neural Information Processing Systems (NeurIPS 2020), Vancouver, Canada.

examples [5, 16]. It uses a loss function that explicitly incorporates Kullback-Leibler (KL)-divergence between the model output and a target Dirichlet with a pre-specified precision value. However, we show that for in-domain examples with high data uncertainties, their proposed loss function distributes the target precision values among the overlapping classes, leading to much flatter distributions. Hence, it often produces indistinguishable representations for such in-domain misclassified examples and OOD examples, compromising the OOD detection performance.

In this work, we propose an alternative approach for a DPN classifier that produces *sharp, multi-modal* Dirichlet distributions for OOD examples to maximize their *representation gap* from in-domain examples. We design a new loss function that separately models the mean and the precision of the output Dirichlet distributions by introducing a novel *explicit precision regularizer* along with the cross-entropy loss. Experimental results on several benchmark datasets demonstrate that our proposed approach achieves the best OOD detection performance.

2 Related Work

In the Bayesian neural network, the predictive uncertainty of a classification model is expressed in terms of data and model uncertainty [4]. Let $\mathcal{D}_{in} = \{\boldsymbol{x}_i, y_i\}_{i=1}^{N} \sim P_{in}(\boldsymbol{x}, y)$ where $\boldsymbol{x}$ and y denotes the images and their corresponding class-labels, sampled from an underlying probability distribution $P_{in}(\boldsymbol{x}, y)$. Given an input $\boldsymbol{x}^*$, the data uncertainty, $p(\omega_c|\boldsymbol{x}^*, \boldsymbol{\theta})$ is the posterior distribution over class labels given the model parameters $\boldsymbol{\theta}$, while the model uncertainty, $p(\boldsymbol{\theta}|\mathcal{D}_{in})$ is the posterior distribution over parameters given the data, $\mathcal{D}_{in}$. Hence, the predictive uncertainty is given as:

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

where ω_c is the representation for class c. We use the standard abbreviation for $p(y = \omega_c|\boldsymbol{x}^*, \mathcal{D}_{in})$ as $p(\omega_c|\boldsymbol{x}^*, \mathcal{D}_{in})$.

However, the true posterior of $p(\boldsymbol{\theta}|\mathcal{D}_{in})$ is intractable. Hence, we need approximation such as Monte-Carlo dropout (MCDP) [11], Langevin Dynamics [17], explicit ensembling [13]: $p(\omega_c|\boldsymbol{x}^*, \mathcal{D}_{in}) \approx \frac{1}{M}\sum_{m=1}^{M} p(\omega_c|\boldsymbol{x}^*, \boldsymbol{\theta}^{(m)})$. where, $\boldsymbol{\theta}^{(m)} \sim q(\boldsymbol{\theta})$ is sampled from an explicit or implicit variational approximation, $q(\boldsymbol{\theta})$ of the true posterior $p(\boldsymbol{\theta}|\mathcal{D}_{in})$. Each $p(\omega_c|\boldsymbol{x}^*, \boldsymbol{\theta}^{(m)})$ represents a categorical distribution, $\boldsymbol{\mu} = [\mu_1, \cdots, \mu_K] = [p(y = \omega_1), \cdots, p(y = \omega_K)]$ over class labels, given $\boldsymbol{x}^*$. Hence, the ensemble can be visualized as a collection of points on the probability simplex. For a confident prediction, it should be appeared sharply in one corner of the simplex. For an OOD example, it should be spread uniformly. We can determine the source of uncertainty in terms of the model uncertainty by measuring their spread. However, producing an ensemble distribution is computationally expensive. Further, it is difficult to control the desired behavior in practice [16]. Furthermore, for standard DNN models, with millions of parameters, it is even harder to find an appropriate prior distribution and the inference scheme to estimate the posterior distribution of the model.

Few recent works, such as Dirichlet prior network (DPN) [5, 16], evidential deep learning (EDL) [18] etc, attempt to emulate this behavior by placing a Dirichlet distribution as a prior, over the predictive categorical distribution. In particular, DPN framework [5, 16] significantly improves the OOD detection performance by explicitly incorporating OOD training examples, $\mathcal{D}_{out}$, as we elaborate in Section 3.

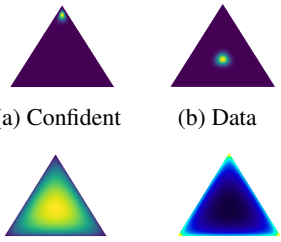
Non-Bayesian frameworks derive their measure of uncertainties using their predictive posteriors obtained from DNNs. Several works demonstrate that tweaking the input images using adversarial perturbations [19] can enhance the performance of a DNN for OOD detection [20, 21]. However, these approaches are sensitive to the tuning of parameters for each OOD distribution and difficult to apply for real-world applications. DeVries & Taylor (2018) [22] propose an auxiliary confidence estimation branch to derive OOD scores. Shalev et al. (2018) [23] use multiple semantic dense representations as the target label to train the OOD detection network. The works in [14, 15] also introduce multi-task loss, incorporating OOD data for training. Hein et al. (2019) [24] show that ReLU-networks lead to over-confident predictions even for samples that are far away from the in-domain distributions and propose methods to mitigate this problem [24, 25, 26]. While these models can identify the total predictive uncertainties, they cannot robustly determine whether the source of uncertainty is due to an in-domain data misclassification or an OOD example.

3 Dirichlet Prior Network

A DPN classification model directly parametrizes a Dirichlet distribution as a prior to the predictive categorical distribution over the probability simplex [5, 16]. It attempts to produce a sharp Dirichlet in a corner when it is confident in its predictions for in-domain examples (Fig 1a). For in-domain examples with high data uncertainty, it attempts to produce a sharp distribution in the middle of the simplex (Fig 1b). Note that, the probability densities for both Dirichlet distributions in Fig 1a and Fig 1b are concentrated in a single mode.

For OOD examples, an existing DPN produces a flat Dirichlet distribution to indicate high-order distributional uncertainty (see Fig 1c). However, we demonstrate that in the case of high data uncertainty among multiple classes, an existing DPN model also produces flatter Dirichlet distributions, leading to indistinguishable representations from OOD examples (see Section 4). Hence, we propose to produce sharp multi-modal Dirichlet distributions for OOD examples to increase their *"representation gap"* from in-domain examples, and improve the OOD detection performance (see Fig 1d). Note that, compared to Fig 1a or Fig 1b, the probability densities of both Dirichlet distributions in Fig 1c and Fig 1d are more scattered over the simplex. We can compute this *"diversity"* using measures, such as *"mutual information (MI)"*, to detect the OOD examples [5]. The predictive uncertainty of a DPN is given as:

Figure 1: Desired behavior of DPN classifiers to indicate different predictive uncertainty types.



(a) Confident (b) Data

(c) Distributional (Existing) (d) Distributional (Proposed)

$$p(\omega_c|\boldsymbol{x}^*, \mathcal{D}) = \int\int p(\omega_c|\boldsymbol{\mu}) p(\boldsymbol{\mu}|\boldsymbol{x}^*, \boldsymbol{\theta}) p(\boldsymbol{\theta}|\mathcal{D}) d\boldsymbol{\mu}\, d\boldsymbol{\theta} \tag{2}$$

where $\mathcal{D}$ denotes the training examples comprising both in-domain, $\mathcal{D}_{in}$, and OOD examples, $\mathcal{D}_{out}$.

Here, the data uncertainty, $p(\omega_c|\boldsymbol{\mu})$ is represented by point-estimate categorical distribution $\boldsymbol{\mu}$, while the distributional uncertainty is represented by using the distribution over the predictive categorical i.e. $p(\boldsymbol{\mu}|\boldsymbol{x}^*, \theta)$. A high model uncertainty, $p(\boldsymbol{\theta}|\mathcal{D})$ would induce a high variation in distributional uncertainty, leading to larger data uncertainty.

DPN is consistent with existing approaches where an additional term is incorporated for distributional uncertainty. Marginalization of $\boldsymbol{\mu}$ in Eqn. 2 produces Eqn. 1. Further, marginalizing $\boldsymbol{\theta}$ produces the expected estimation of data and distributional uncertainty given model uncertainty, i.e,

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

However, as in Eqn. 1, marginalization of $\boldsymbol{\theta}$ is not tractable. Hence, as before, we can introduce approximation techniques for $p(\boldsymbol{\theta}|\mathcal{D})$ to measure model uncertainty [11, 13]. However, the model uncertainty is *reducible* given large amount of training data [4]. Here, we only focus on the uncertainties introduced by the input examples and assume a dirac-delta approximation $\hat{\boldsymbol{\theta}}$ for $\boldsymbol{\theta}$ for the DPN models [5]: $p(\boldsymbol{\theta}|\mathcal{D}) = \delta(\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}) \implies p(\boldsymbol{\mu}|\boldsymbol{x}^*, \mathcal{D}) \approx p(\boldsymbol{\mu}|\boldsymbol{x}^*, \hat{\boldsymbol{\theta}})$.

Construction. A Dirichlet distribution is parameterized using a vector, called the concentration parameters, $\boldsymbol{\alpha} = \{\alpha_1, \cdots, \alpha_K\}$, as: $Dir(\boldsymbol{\mu}|\boldsymbol{\alpha}) = \frac{\Gamma(\alpha_0)}{\prod_{c=1}^K \Gamma(\alpha_c)} \prod_{c=1}^K \mu_c^{\alpha_c - 1}, \quad \alpha_c > 0$, where, $\alpha_0 = \sum_{c=1}^K \alpha_c$) denotes its precision. A larger precision leads to a sharper Dirichlet distribution.

A DPN, $f_{\hat{\theta}}$ produces the concentration parameters, $\boldsymbol{\alpha} = \{\alpha_1, \cdots, \alpha_K\}$ corresponding to each class, i.e, $\boldsymbol{\alpha} = f_{\hat{\theta}}(\boldsymbol{x}^*)$. The posterior over class labels is given by the mean of the Dirichlet, i.e,

$$p(\omega_c|\boldsymbol{x}^*; \hat{\boldsymbol{\theta}}) = \int p(\omega_c|\boldsymbol{\mu})\ p(\boldsymbol{\mu}|\boldsymbol{x}^*; \hat{\boldsymbol{\theta}})\ d\boldsymbol{\mu} = \frac{\alpha_c}{\alpha_0}, \qquad \text{where,}\ \ p(\boldsymbol{\mu}|\boldsymbol{x}^*; \hat{\theta}) = Dir(\boldsymbol{\mu}|\boldsymbol{\alpha}) \tag{4}$$

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:

$$p(\omega_c|\boldsymbol{x}^*; \hat{\boldsymbol{\theta}}) = \frac{\alpha_c}{\alpha_0} = \frac{e^{z_c(\boldsymbol{x}^*)}}{\sum_{c=1}^K e^{z_c(\boldsymbol{x}^*)}} \tag{5}$$

However, the mean of the Dirichlet distribution is now *insensitive* to any arbitrary scaling of α_c. Hence, while the standard *cross-entropy loss* efficiently models the mean of the Dirichlet distributions, it degrades the precision, α_0.

Malinin & Gales (2018) [5] propose a forward KL (FKL) divergence loss that explicitly minimizes the KL divergence between the model and the given target Dirichlet distribution. Malinin & Gales (2019) [16] further propose a reverse KL (RKL) loss function that reverses the terms in the KL divergence to induce a uni-modal Dirichlet as the target distribution and improve their scalability for classification tasks with a larger number of classes. The RKL loss trains a DPN using both in-domain and OOD training examples in a multi-task fashion:

$$\mathcal{L}^{rkl}(\theta;\gamma,\boldsymbol{\beta}^{y},\boldsymbol{\beta}^{out}) = \mathbb{E}_{P_{in}}\text{KL}[p(\boldsymbol{\mu}|\boldsymbol{x},\boldsymbol{\theta})||Dir(\boldsymbol{\mu}|\boldsymbol{\beta}^{y})] + \gamma \cdot \mathbb{E}_{P_{out}}\text{KL}[p(\boldsymbol{\mu}|\boldsymbol{x},\boldsymbol{\theta})||Dir(\boldsymbol{\mu}|\boldsymbol{\beta}^{out})] \quad (6)$$

where P_{in} and P_{out} are the distribution for the in-domain and OOD training examples and $\boldsymbol{\beta}^{y}$ and $\boldsymbol{\beta}^{out}$ their hand-crafted target concentration parameters respectively.

4 Proposed Methodology

Shortcomings of DPN using RKL loss. We first demonstrate that the RKL loss function tends to produce flatter Dirichlet distributions for in-domain misclassified examples, compared to its confident predictions. We can decompose the reverse KL-divergence loss into two terms i.e *reverse cross entropy*, $\mathbb{E}_{P(\boldsymbol{\mu}|\boldsymbol{x},\boldsymbol{\theta})}[-\ln Dir(\boldsymbol{\mu}|\overline{\boldsymbol{\beta}})]$ and *differential entropy*, $\mathcal{H}[p(\boldsymbol{\mu}|\boldsymbol{x},\boldsymbol{\theta})]$, as shown in [16]:

$$\mathbb{E}_{\tilde{P}_T(\boldsymbol{x},y)}\,\text{KL}\big[\,p(\boldsymbol{\mu}|\boldsymbol{x},\boldsymbol{\theta})\;||\;Dir(\boldsymbol{\mu}|\boldsymbol{\beta})\,\big] = \mathbb{E}_{\tilde{P}_T(\boldsymbol{x})}\Big[\mathbb{E}_{P(\boldsymbol{\mu}|\boldsymbol{x},\boldsymbol{\theta})}[-\ln Dir(\boldsymbol{\mu}|\overline{\boldsymbol{\beta}})] - \mathcal{H}\big[p(\boldsymbol{\mu}|\boldsymbol{x},\boldsymbol{\theta})\big]\Big] \quad (7)$$

where $\boldsymbol{\beta} = \{\beta_1^{(c)},\cdots,\beta_K^{(c)}\}$ represents their hand-crafted target concentration parameters, and $\overline{\boldsymbol{\beta}}$ represents the concentration parameter of the expected target Dirichlet with respect to the empirical training distribution $\tilde{P}_T$. In our analysis, we replace $\tilde{P}_T$ with the empirical distribution of in-domain training examples $\tilde{P}_{in}$ or OOD training examples $\tilde{P}_{out}$.

Differential entropy measures the sharpness of a continuous distribution. Minimizing $-\mathcal{H}[p(\boldsymbol{\mu}|\boldsymbol{x},\boldsymbol{\theta})]$ always leads to produce a flatter distribution. Hence, we rely only on $\mathbb{E}_{P(\boldsymbol{\mu}|\boldsymbol{x},\boldsymbol{\theta})}[-\ln Dir(\boldsymbol{\mu}|\overline{\boldsymbol{\beta}})]$ to produce sharper distributions. Malinin & Gales (2019) [16] choose the target concentration value for in-domain examples as: $(\beta+1)$ for the correct class and 1 for the incorrect classes. Thus, we get:

$$\begin{aligned}\mathbb{E}_{\tilde{P}_T(\boldsymbol{x})}\Big[\mathbb{E}_{P(\boldsymbol{\mu}|\boldsymbol{x},\boldsymbol{\theta})}[-\ln Dir(\boldsymbol{\mu}|\overline{\boldsymbol{\beta}})]\Big] &= \mathbb{E}_{\tilde{P}_T(\boldsymbol{x})}\Big[\sum_c\sum_k \tilde{p}(\omega_c|\boldsymbol{x})(\beta_k^{(c)}-1)\big[\psi(\alpha_0)-\psi(\alpha_k)\big]\Big] \\ &= \mathbb{E}_{\tilde{P}_T(\boldsymbol{x})}\Big[\beta\psi(\alpha_0) - \sum_c \beta\tilde{p}(\omega_c|\boldsymbol{x})\psi(\alpha_c)\Big]\end{aligned} \quad (8)$$

where ψ is the digamma function.

We can see in Eqn. 8, the reverse cross-entropy term maximizes $\psi(\alpha_c)$ for each class c with the factor, $\beta\tilde{p}(\omega_c|\boldsymbol{x})$, and minimizes $\psi(\alpha_0)$ with the factor, β. For an in-domain example with confident prediction, it produces a sharp Dirichlet with a large concentration value for the correct class and very small concentration parameters ($<< 1$) for the incorrect classes. However, for an input with high data uncertainty, β is distributed among multiple classes according to $\tilde{p}(\omega_c|\boldsymbol{x})$. This leads to relatively smaller (but ≥ 1) concentration parameters for all overlapping classes, producing a much flatter and diverse Dirichlet.

Figure 2: Dirichlet distributions with the same precision but different concentration parameters.

(a) Dir_1:$\{0.01, 0.01, 101.98\}$ D.Ent = -199.1, MI=$8e$-4

(b) Dir_2:$\{34, 34, 34\}$ D.Ent = -3.45, MI=$9.7e$-3

For example, let us consider two Dirichlet distributions, Dir_1 and Dir_2, with the same precision, $\alpha_0 = 102$, but different concentration parameters of $\{0.01, 0.01, 101.98\}$ (Fig. 2a) and $\{34, 34, 34\}$ (Fig. 2b). We measure the differential entropy (D.Ent) and mutual information (MI), which are maximized for flatter and diverse distributions respectively. We can see that Dir_1 produces much lower D.Ent and MI scores than Dir_2. It shows that Dir_2 is flatter and diverse than Dir_1, even with the same precision values. The differences in these scores become more significant in higher dimensions. As we consider the same example for a classification task with $K = 100$, D.Ent and MI would respectively produce $-9.9e3$ and 0.02 for Dir_1 and -370.5 and 0.23 for Dir_2. Further, in Section 4 (and in Table 9 (Appendix)), we show that DPN models also tend to produce lower precision values along with flatter and diverse Dirichlet distributions for misclassified examples.

This behavior is *not* desirable: since the RKL loss also trains the DPN to produce flatter and diverse Dirichlet distributions for OOD examples, it often leads to indistinguishable distributions for OOD examples in the boundary cases and in-domain misclassified examples. However, we can produce sharper Dirichlet distributions for OOD examples, as in Figure 1d, to maximize their representation gap from in-domain examples. For OOD training examples, we should choose identical values for target concentration parameters, $(\tau+1)$ where $\tau > -1$, for all classes to produce uniform categorical posterior distributions. Using $(\tau+1)$ for $\beta_k^{(c)}$ in Eqn. 8, we get the RKL loss for OOD examples as:

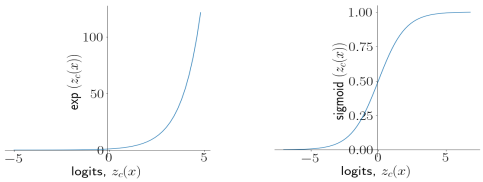
$$\mathbb{E}_{P_T(\boldsymbol{x})}\Big[\tau K\psi(\alpha_0) - \sum_c \tau\psi(\alpha_c) - \mathcal{H}\big[p(\boldsymbol{\mu}|\boldsymbol{x},\boldsymbol{\theta})\big]\Big] \tag{9}$$

Malinin & Gales (2019) [16] choose τ to 0. This results the RKL loss to minimize $-\mathcal{H}[p(\boldsymbol{\mu}|\boldsymbol{x},\boldsymbol{\theta})]$. Hence, the DPN produces flat Dirichlet distributions for OOD examples. In the following, we investigate the other choices of τ.

Choosing $\tau > 0$ gives an objective function that minimizes the precision, $\alpha_0 (= \sum_{c=1}^{K} \alpha_c)$ of the output Dirichlet distribution while maximizing individual concentration parameters, α_c (Eq. 9). In contrast, choosing $\tau \in (-1, 0)$ maximizes α_0 while minimizing α_c's. Hence, we can conclude that either choice of τ may lead to *uncontrolled values* for the concentration parameters for an OOD example. We also empirically verify this analysis in Appendix A.2.

Explicit Precision Regularization. We propose a new loss function for DPN classifiers that separately models the mean and precision of the output Dirichlet to achieve greater control over the desired behavior of a DPN classifier. We model the mean of the output Dirichlet using soft-max activation function and the cross-entropy loss, as in Eqn. 5, along with a novel *explicit precision regularizer*, that controls the precision.

Figure 3: Growth of regularizers w.r.t logits.



(a) $\sum_c \exp z_c(\boldsymbol{x})$ (b) $\frac{1}{K}\sum_c \text{sigmoid} z_c(\boldsymbol{x})$

Note that, we cannot choose the regularizer that directly maximizes the precision, $\alpha_0 = \sum_{c=1}^{K} \exp z_c(\boldsymbol{x})$ to control the output Dirichlet distributions. This is because the term, $\exp z_c(\boldsymbol{x})$ is unbounded. Hence, using the precision, $\alpha_0 = \sum_c \exp z_c(\boldsymbol{x})$ as the regularizer leads to large logit values for in-domain examples (see Figure 3(a)). However, it would make the cross-entropy loss term negligible, degrading the in-domain classification accuracy. Further, $\exp z_c(\boldsymbol{x})$ is not a symmetric function. Hence, it does not equally constrain the network to produce small fractional concentration parameters, i.e $\alpha_c = \exp z_c(\boldsymbol{x}) \to 0$, for OOD examples to produce the desirable multi-modal Dirichlet distributions (Figure 1d). Moreover, in practice, the choice of $\sum_c \exp z_c(\boldsymbol{x})$ leads the training loss to NaN.

In contrast, by limiting logits, z_c to values that are, for example, approximately 5 for in-domain examples, and -5 for OOD examples, we would have the desirable sharp uni-modal and multi-modal Dirichlet distributions respectively, maximizing their representation gaps (see Figure 1). Beyond these values, the cross-entropy loss should become the dominant term in the loss function to improved the in-domain classification accuracy.

Hence, we propose a logistic-sigmoid approximation to *individually control* the concentration parameters using $\frac{1}{K}\sum_{c=1}^{K} \text{sigmoid}(z_c(\boldsymbol{x}))$ as the regularizer to control the spread of the output Dirichlet distributions. This regularizer is applied alongside the cross-entropy loss on the soft-max outputs. The use of logistic-sigmoid function satisfies this condition by providing an implicit upper and lower bounds on the individual concentration parameters for both in-domain and OOD examples (see Figure 3(b)). For interval $(-\infty, \epsilon)$, when ϵ is close to 0, the approximation error is low. While for the interval $[\epsilon, \infty)$, it offers the desired behavior of monotonically increasing function, similar to the exponential function, however within a finite boundary.

The proposed loss function for the in-domain training examples is given as:

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

Similarly, for OOD training examples, we can control precision values using the loss function as:

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

where $\mathcal{H}_{ce}$ is the cross-entropy function. $\mathcal{U}$ is the uniform distribution over the class labels. λ_{in} and λ_{out} are user-defined hyper-parameters for the regularization terms to control the precision of the output distributions. We train the DPN in a multi-task fashion using the overall loss function as:

$$\min_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}; \gamma, \lambda_{in}, \lambda_{out}) = \mathcal{L}_{in}(\boldsymbol{\theta}, \lambda_{in}) + \gamma \mathcal{L}_{out}(\boldsymbol{\theta}, \lambda_{out}) \tag{12}$$

where $\gamma > 0$ balances between the loss values for in-domain examples and OOD examples. We now analyze the proposed regularizer by taking expectation with respect to the empirical distribution, $\tilde{P}_T$:

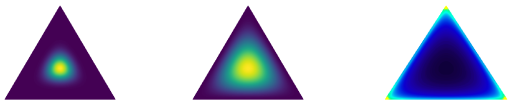
$$\begin{aligned} \mathbb{E}_{\tilde{P}_T(\boldsymbol{x},y)}\Big[-\frac{\lambda_T}{K}\sum_{c=1}^{K} \text{sigmoid}(z_c(\boldsymbol{x}))\Big] &= \mathbb{E}_{P_T(\boldsymbol{x})}\Big[-\frac{\lambda_T}{K}\sum_{k=1}^{K} p(y=\omega_k|\boldsymbol{x})\Big[\sum_{c=1}^{K} \text{sigmoid}(z_c(\boldsymbol{x}))\Big]\Big] \\ &= \mathbb{E}_{P_T(\boldsymbol{x})}\Big[-\frac{\lambda_T}{K}\sum_{c=1}^{K} \text{sigmoid}(z_c(\boldsymbol{x}))\Big] \end{aligned} \tag{13}$$

where λ_T and $\tilde{P}_T$ should be replaced to λ_{in} and $\tilde{P}_{in}$ respectively for in-domain examples, and λ_{out} and $\tilde{P}_{out}$ respectively for OOD examples.

Hence, by choosing $\lambda_{in} > 0$ for in-domain examples, our regularizer imposes the network to maximize $\text{sigmoid}(z_c(\boldsymbol{x}))$ irrespective of the class-labels. However, for confidently predicted examples, the cross-entropy loss ensures to maximize the logit value for the correct class. In contrast, in the presence of high data uncertainty, the cross-entropy loss produces a multi-modal categorical distribution over the overlapping classes. Hence, as before, it leads to producing a flatter distribution for misclassified examples. Now, by choosing $\lambda_{in} > \lambda_{out} > 0$, we also enforce the network to produce a flatter distribution with $\alpha_c = \exp z_c(\boldsymbol{x}^*) \geq 1$ for an OOD example $\boldsymbol{x}^*$. Hence, the DPN will produce indistinguishable representations for an in-domain example with high data uncertainty as an OOD example, similar to the RKL loss (Eq. 7).

However, now we can address this problem by choosing $\lambda_{out} < 0$. It enforces the DPN to produce negative values for $z_c(\boldsymbol{x}^*)$ and thus *fractional* values for α_c's for OOD examples. This leads the probability densities to be moved across the edges of the simplex to produce extremely sharp multi-modal distributions, solving the original problem described in the beginning.

Figure 4: Dirichlet distributions with different precision.



(a) $\boldsymbol{\alpha}$= {10, 10, 10} D.Ent=-2.3 MI=0.03 (b) $\boldsymbol{\alpha}$= (3, 3, 3) D.Ent=-1.2 MI=0.1 (c) $\boldsymbol{\alpha}$= (0.1, 0.1, 0.1) D.Ent=-13.0, MI=0.84

For example, let a DPN with $\lambda_{in}, \lambda_{out} > 0$ represents a misclassified example (Fig. 4a) and an OOD example (Fig. 4b). We can see that their representations are very similar, even if their concentration parameters are different. In contrast, our DPN with $\lambda_{in} > 0, \lambda_{out} < 0$ leads to a sharp, multi-modal Dirichlet as in Fig. 4(c) for an OOD, maximizing the *representation gap* from Fig. 4(a). We can confirm this by observing their *mutual information (MI)* scores. However, the choice of $\lambda_{in} = 0$ and $\lambda_{out} < 0$ does not enforce these properties (see ablation study in Appendix A.1).

The overall loss function in Eqn. 12 requires training samples from both in-domain distribution and OOD. Here, we select a different real-world dataset as our OOD training examples. It is more feasible in practice and performs better than the artificially generated OOD examples [15, 14].

5 Experimental Study

We conduct two sets of experiments: First, we experiment on a synthetic dataset. Next, we present a comparative study on a range of image classification tasks. [1] [2]

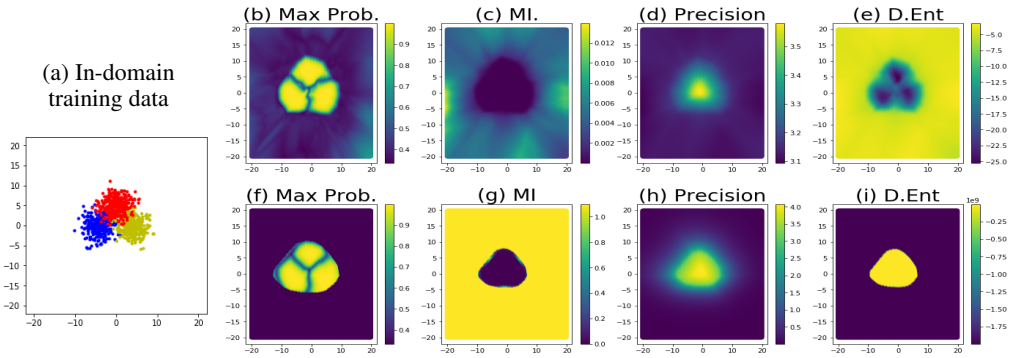
5.1 Synthetic Dataset

We construct a simple synthetic dataset with three overlapping classes to study the characteristics of different DPN models. We sample the in-domain training instances from three different overlapping

[1] Please refer to Appendix for additional results and ablation studies.

[2] Code Link: https://github.com/jayjaynandy/maximize-representation-gap

Figure 5: Uncertainty measures of different data-points for DPN$^+$ (top row) and DPN$^-$ (bottom row).



isotropic Gaussian distributions, to obtain these overlapping classes, as shown in Figure 5a. We demonstrate the results of our DPN models with $\lambda_{in} > 0$ and both positive and negative values for λ_{out}, are denoted as DPN$^+$ and DPN$^-$ respectively. See Appendix B.1 for additional details on experimental setup, hyper-parameters and the results using RKL loss function [16].

A *total predictive uncertainty* measure is derived from the expected predictive categorical distribution, $p(\omega_c|\boldsymbol{x}^*, D)$ i.e by marginalizing $\boldsymbol{\mu}$ and $\boldsymbol{\theta}$ in Eq. 2. Maximum probability in the expected predictive categorical distribution is a popular total uncertainty measure: $max\mathcal{P} = \max_c p(\omega_c|\boldsymbol{x}^*, D)$.

Fig 5b and Fig 5f show the Max.P uncertainty measures for different data points for DPN$^+$ and DPN$^-$ respectively. We can see that DPN$^-$ appropriately interpolates the concentration parameters to maximize the margin on the boundary of the in-domain and OOD regions, leading to improved OOD detection performance even using total uncertainty measures. However, since the predicted distributions are obtained by marginalizing $\boldsymbol{\mu}$ (Eqn. 2), the total uncertainty measures fail to robustly distinguish the OOD examples from the misclassified examples. As we can see in both Fig 5b and Fig 5f that the Max.P produces lower scores in the class overlapping regions. Since the non-Bayesian models only relies on the total uncertainty measures, they are unable to reliably distinguish between data and distributional uncertainty [5].

A DPN can address this limitation by computing the *mutual information (MI)* between y and $\boldsymbol{\mu}$ i.e $\mathcal{I}[y, \boldsymbol{\mu}|\boldsymbol{x}^*, \hat{\boldsymbol{\theta}}]$. We can also measure the *expected pairwise KL divergence (EPKL)* between the pairs of independent "categorical distribution" samples from the Dirichlet [27]. For a Dirichlet distribution, EPKL is simplified to $\frac{K-1}{\alpha_0}$, where α_0 is the precision [27]. Since a DPN also produces smaller precision values for OOD examples, we can directly view the *precision* as a distributional uncertainty measure. Note that, both EPKL and *precision (or inverse-EPKL)* leads to the same OOD detection performance as they produce the same relative uncertainty scores (in reverse order) for a given set of test examples. In Fig 5c and Fig 5d, we observe that our DPN$^+$ model successfully distinguishes the OOD examples using the mutual information and the precision measures respectively. However, our DPN$^-$ model clearly demonstrates its superiority by producing sharper and significant differences of uncertainty scores for OOD examples, compared to the in-domain examples (Fig 5g and Fig 5h).

Differential entropy (D.Ent), $\mathcal{H}[p(\boldsymbol{\mu}|\boldsymbol{x}^*, \hat{\theta})]$, that maximizes for sharp Dirichlet distributions, is also used as a distributional uncertainty measure [5, 27]. However, unlike other DPN models, our DPN$^-$ behaves differently to produce sharp multi-modal Dirichlet distributions for OOD examples. Hence, D.Ent also behaves in an *inverted manner*, compared to the other DPN models. As we can see in Fig 5e, DPN$^+$ produces higher D.Ent values for OOD examples. In contrast, DPN$^-$ produces *large negative* D.Ent scores for OOD examples, indicating that it often produces even sharper Dirichlet for OOD examples, than the confidently predicted examples (Fig 5i).

5.2 Benchmark Image Classification Datasets

Next, we carry out experiments on CIFAR-10 and CIFAR-100 [28] and TinyImageNet [29]. We train the C10 classifiers by using CIFAR-10 training images as in-domain data and CIFAR-100 training images as OOD data. C100 classifiers are trained by using CIFAR-100 training images as in-domain and CIFAR-10 training images as OOD. For the TIM classifier, we use the TinyImageNet images as in-domain training data and ImageNet-25K images as OOD training data. ImageNet-25K is obtained

by randomly selecting $25,000$ images from the ImageNet dataset [30]. We use the VGG-16 network for these tasks [1]. Here, we study the performance our DPN models with $\lambda_{in} > 0$ and both $\lambda_{out} > 0$ and $\lambda_{out} < 0$, are denoted as DPN$^+$ and DPN$^-$ respectively. See Appendix A for additional ablation studies. We compare our models with the standard DNN [31], Bayesian MCDP [11], Deep Ensemble (DE) [13], non-Bayesian OE [15] and the existing DPN$_{rev}$ model [16]. For Bayesian MCDP and DE, we can compute the mutual information (MI). However, we cannot compute the precision or D.Ent for them. For the non-Bayesian models, MI, precision, and D.Ent are not defined. See Appendix B.2 for our experimental details along with additional discussions.

We present the performances of our models for *OOD detection* and *misclassification detection* tasks. Note that the in-domain and OOD test examples are kept separately from the training examples, as in a real-world scenario (see Table 7(Appendix)). For OOD detection, we choose the OOD examples as the 'positive' class and in-domain examples as the 'negative' class. For misclassification detection, we consider the misclassified examples as the 'positive' class and correctly classified examples as the 'negative' class. Here, we use *area under the receiver operating characteristic (AUROC)* metric [31]. We present the results using Max.P, MI, α_0 (or inverse-EPKL), and D.Ent. We report the (mean $\pm$ standard deviation) of three different models. We provide additional results including classification accuracy, and performance on a wide range of OOD datasets along with area under the precision-recall curve (AUPR) metric and entropy measure in Appendix D (Table 9-13).

Table 1: AUROC scores for OOD detection (mean $\pm$ standard deviation of 3 runs). Refer to Table 11-13 (Appendix) for AUPR scores and results on additional OOD test sets.

	OOD	Tiny [29]				STL-10 [32]				LSUN [33]			
		Max.P	MI	α_0	D.Ent	Max.P	MI	α_0	D.Ent	Max.P	MI	α_0	D.Ent
C10	Baseline	88.9$_{\pm 0.0}$	-	-	-	75.9$_{\pm 0.0}$	-	-	-	90.3$_{\pm 0.0}$	-	-	-
	MCDP	88.7$_{\pm 0.1}$	88.1$_{\pm 0.1}$	-	-	76.2$_{\pm 0.0}$	76.0$_{\pm 0.0}$	-	-	90.6$_{\pm 0.0}$	90.2$_{\pm 0.0}$	-	-
	DE	88.9$_{\pm \text{NA}}$	87.8$_{\pm \text{NA}}$	-	-	76.0$_{\pm \text{NA}}$	75.6$_{\pm \text{NA}}$	-	-	90.3$_{\pm \text{NA}}$	89.7$_{\pm \text{NA}}$	-	-
	OE	98.2$_{\pm 0.1}$	-	-	-	81.4$_{\pm 1.2}$	-	-	-	98.4$_{\pm 0.3}$	-	-	-
	DPN$_{rev}$	97.5$_{\pm 0.5}$	97.8$_{\pm 0.4}$	97.8$_{\pm 0.4}$	97.7$_{\pm 0.4}$	81.6$_{\pm 1.7}$	82.2$_{\pm 1.7}$	82.2$_{\pm 1.6}$	81.9$_{\pm 1.7}$	98.5$_{\pm 0.4}$	98.7$_{\pm 0.3}$	98.7$_{\pm 0.3}$	98.7$_{\pm 0.3}$
	DPN$^+$	98.0$_{\pm 0.2}$	98.0$_{\pm 0.2}$	98.0$_{\pm 0.2}$	98.0$_{\pm 0.2}$	81.6$_{\pm 1.4}$	81.8$_{\pm 1.2}$	81.8$_{\pm 1.2}$	81.8$_{\pm 1.2}$	98.2$_{\pm 0.3}$	98.3$_{\pm 0.4}$	98.3$_{\pm 0.4}$	98.3$_{\pm 0.4}$
	DPN$^-$	**99.0$_{\pm 0.1}$**	**99.0$_{\pm 0.1}$**	97.7$_{\pm 0.1}$	6.0$_{\pm 0.3}$	84.7$_{\pm 0.4}$	**85.3$_{\pm 0.5}$**	84.9$_{\pm 0.5}$	34.6$_{\pm 0.4}$	99.2$_{\pm 0.1}$	**99.3$_{\pm 0.0}$**	98.1$_{\pm 0.1}$	5.0$_{\pm 0.2}$
	OOD	Tiny [29]				STL-10 [32]				LSUN [33]			
		Max.P	MI	α_0	D.Ent	Max.P	MI	α_0	D.Ent	Max.P	MI	α_0	D.Ent
C100	Baseline	68.8$_{\pm 0.2}$	-	-	-	69.6$_{\pm 0.0}$	-	-	-	72.5$_{\pm 0.0}$	-	-	-
	MCDP	69.7$_{\pm 0.3}$	70.6$_{\pm 0.3}$	-	-	70.7$_{\pm 0.1}$	71.6$_{\pm 0.2}$	-	-	74.5$_{\pm 0.1}$	75.9$_{\pm 0.2}$	-	-
	DE	68.9$_{\pm \text{NA}}$	69.6$_{\pm \text{NA}}$	-	-	69.6$_{\pm \text{NA}}$	70.2$_{\pm \text{NA}}$	-	-	72.6$_{\pm \text{NA}}$	73.4$_{\pm \text{NA}}$	-	-
	OE	89.5$_{\pm 1.0}$	-	-	-	91.2$_{\pm 0.7}$	-	-	-	92.2$_{\pm 0.9}$	-	-	-
	DPN$_{rev}$	81.2$_{\pm 0.2}$	83.8$_{\pm 0.1}$	83.8$_{\pm 0.1}$	83.5$_{\pm 0.1}$	87.2$_{\pm 0.1}$	89.3$_{\pm 0.1}$	89.3$_{\pm 0.1}$	89.0$_{\pm 0.1}$	86.7$_{\pm 0.0}$	89.3$_{\pm 0.1}$	89.3$_{\pm 0.1}$	88.9$_{\pm 0.1}$
	DPN$^+$	85.9$_{\pm 0.3}$	92.2$_{\pm 0.1}$	92.2$_{\pm 0.1}$	92.3$_{\pm 0.1}$	89.1$_{\pm 0.2}$	95.0$_{\pm 0.0}$	95.0$_{\pm 0.0}$	94.8$_{\pm 0.0}$	90.3$_{\pm 0.3}$	95.0$_{\pm 0.1}$	95.0$_{\pm 0.1}$	95.0$_{\pm 0.1}$
	DPN$^-$	89.2$_{\pm 0.1}$	**94.5$_{\pm 0.1}$**	**94.5$_{\pm 0.1}$**	38.1$_{\pm 0.5}$	92.8$_{\pm 0.1}$	**96.8$_{\pm 0.1}$**	**96.8$_{\pm 0.1}$**	25.4$_{\pm 0.4}$	92.8$_{\pm 0.1}$	**96.5$_{\pm 0.1}$**	**96.5$_{\pm 0.1}$**	31.5$_{\pm 0.4}$
	OOD	CIFAR-10 [28]				CIFAR-100 [28]				Textures [34]			
		Max.P	MI	α_0	D.Ent	Max.P	MI	α_0	D.Ent	Max.P	MI	α_0	D.Ent
TIM	Baseline	76.9$_{\pm 0.2}$	-	-	-	73.6$_{\pm 0.2}$	-	-	-	70.9$_{\pm 0.2}$	-	-	-
	MCDP	77.4$_{\pm 0.1}$	77.5$_{\pm 0.2}$	-	-	74.0$_{\pm 0.2}$	73.6$_{\pm 0.2}$	-	-	70.3$_{\pm 0.2}$	63.6$_{\pm 0.2}$	-	-
	DE	76.9$_{\pm \text{NA}}$	77.7$_{\pm \text{NA}}$	-	-	73.7$_{\pm \text{NA}}$	75.3$_{\pm \text{NA}}$	-	-	71.1$_{\pm \text{NA}}$	76.2$_{\pm \text{NA}}$	-	-
	OE	91.3$_{\pm 0.4}$	-	-	-	89.5$_{\pm 0.5}$	-	-	-	95.8$_{\pm 0.3}$	-	-	-
	DPN$_{rev}$	85.4$_{\pm 0.7}$	82.8$_{\pm 1.4}$	81.9$_{\pm 1.6}$	85.6$_{\pm 0.9}$	84.2$_{\pm 0.8}$	82.5$_{\pm 1.4}$	81.7$_{\pm 1.6}$	85.0$_{\pm 0.9}$	90.9$_{\pm 0.3}$	91.2$_{\pm 0.6}$	90.6$_{\pm 0.6}$	92.6$_{\pm 0.3}$
	DPN$^+$	99.2$_{\pm 0.0}$	99.7$_{\pm 0.0}$	99.7$_{\pm 0.0}$	99.6$_{\pm 0.0}$	98.8$_{\pm 0.0}$	99.5$_{\pm 0.0}$	99.5$_{\pm 0.0}$	99.4$_{\pm 0.0}$	96.5$_{\pm 0.1}$	98.4$_{\pm 0.0}$	98.4$_{\pm 0.0}$	98.2$_{\pm 0.0}$
	DPN$^-$	99.7$_{\pm 0.0}$	**99.9$_{\pm 0.0}$**	**99.9$_{\pm 0.0}$**	3.5$_{\pm 0.1}$	98.7$_{\pm 0.1}$	**99.6$_{\pm 0.0}$**	**99.6$_{\pm 0.0}$**	7.5$_{\pm 0.2}$	95.8$_{\pm 0.1}$	**98.7$_{\pm 0.1}$**	**98.7$_{\pm 0.1}$**	19.3$_{\pm 0.4}$

Tables 1 shows the performance of C10, C100 and TIM classifiers for *OOD detection* task. We observe that our DPN$^-$ models consistently outperform the other models using mutual information (MI) measure. Our DPN$^-$ models produce sharp multi-modal Dirichlet distributions for OOD examples, leading to higher MI scores compared to the in-domain examples. In contrast, DPN$^-$ models produce sharp Dirichlet distributions for both in-domain confident predictions and OOD examples. Hence, we cannot use D.Ent to distinguish them. However, in Table 3, we show that we can combine D.Ent with a total uncertainty measure to distinguish the in-domain and OOD examples.

Table 2: AUROC scores for misclassified image detection. Refer to Table 9 (Appendix) for additional results by using AUPR scores along with in-domain classification accuracy.

	C10				C100				TIM			
	Max.P	MI	α_0	D.Ent	Max.P	MI	α_0	D.Ent	Max.P	MI	α_0	D.Ent
Baseline	93.3$_{\pm 0.1}$	-	-	-	86.8$_{\pm 0.1}$	-	-	-	86.7$_{\pm 0.0}$	-	-	-
MCDP	**93.6$_{\pm 0.2}$**	93.2$_{\pm 0.1}$	-	-	**87.2$_{\pm 0.0}$**	83.3$_{\pm 0.3}$	-	-	86.6$_{\pm 0.1}$	83.3$_{\pm 0.3}$	-	-
DE	93.5$_{\pm \text{NA}}$	92.7$_{\pm \text{NA}}$	-	-	87.0$_{\pm \text{NA}}$	83.4$_{\pm \text{NA}}$	-	-	**86.8$_{\pm \text{NA}}$**	83.3$_{\pm \text{NA}}$	-	-
OE	92.0$_{\pm 0.0}$	-	-	-	86.9$_{\pm 0.0}$	-	-	-	85.9$_{\pm 0.2}$	-	-	-
DPN$_{rev}$	89.6$_{\pm 0.1}$	88.7$_{\pm 0.2}$	88.7$_{\pm 0.2}$	89.0$_{\pm 0.2}$	79.3$_{\pm 0.1}$	73.5$_{\pm 0.1}$	73.1$_{\pm 0.1}$	75.7$_{\pm 0.1}$	81.9$_{\pm 0.3}$	72.2$_{\pm 0.7}$	70.2$_{\pm 0.9}$	78.3$_{\pm 0.3}$
DPN$^+$	92.2$_{\pm 0.3}$	90.3$_{\pm 0.1}$	90.3$_{\pm 0.1}$	90.5$_{\pm 0.2}$	86.5$_{\pm 0.1}$	81.2$_{\pm 0.0}$	81.3$_{\pm 0.0}$	81.9$_{\pm 0.1}$	85.7$_{\pm 0.2}$	78.3$_{\pm 0.4}$	78.7$_{\pm 0.5}$	79.7$_{\pm 0.2}$
DPN$^-$	92.6$_{\pm 0.1}$	89.9$_{\pm 0.0}$	89.9$_{\pm 0.0}$	66.2$_{\pm 0.7}$	86.4$_{\pm 0.1}$	82.3$_{\pm 0.0}$	82.3$_{\pm 0.0}$	81.7$_{\pm 0.1}$	85.4$_{\pm 0.1}$	79.1$_{\pm 0.5}$	79.4$_{\pm 0.4}$	79.9$_{\pm 0.2}$

Table 2 presents the results for in-domain *misclassification detection*. We can see that our proposed DPN$^-$ models achieve comparable performance as the existing methods. It is interesting to note that, all the DPN models produce comparable AUROC scores using the distributional uncertainty measures as the total uncertainty measure. This supports our assertion that *in the presence of data uncertainty, DPN models tend to produce flatter and diverse Dirichlet distributions with smaller precisions for misclassified examples, compared to the confident predictions.* Hence, we should aim to produce sharp multi-modal Dirichlet distributions for OOD examples to keep them distinguishable from the in-domain examples. In Appendix B.3, we also demonstrate that our DPN$^-$ models improve calibration performance for classification [35, 15].

AUROC (and also AUPR) scores, in Table 1, only provide a relative measure of separation, while providing no information about how different these uncertainty values are for in-domain and OOD examples [36]. An OOD detection model should aim to maximize the margin of uncertainty values produced for the OOD examples from the in-domain examples to separate them efficiently. We can measure the "gap" of uncertainty values for in-domain and OOD examples by measuring the divergence of their distributions for uncertainty values produced by the models.

DPN$^-$ models produce sharp Dirichlet distributions for both OOD examples and in-domain confident predictions, while flat Dirichlet distributions for misclassified examples. Hence, we can combine D.Ent along with a total uncertainty measure, Max.P, to robustly distinguish them. Note that D.Ent ranges from $(-\infty, \infty)$ and often produce large negative values. Hence, we use $\log$(-D.Ent). We separately consider Max.P and $\log$(-D.Ent) scores for the in-domain correctly classified, misclassified, and OOD examples. We fit three different bivariate Gaussian distributions on these values. We then compute the KL-divergence of the distributions of uncertainty values for in-domain confidently predicted and misclassified examples to the OOD examples to measure their separability. Figure 6 illustrates the desired behavior of our DPN$^-$, compared to the other DPN models.

Figure 6: Illustrating the distribution of uncertainty values for DPN$^-$ and other DPN models. We normalize the scores for better visualization.

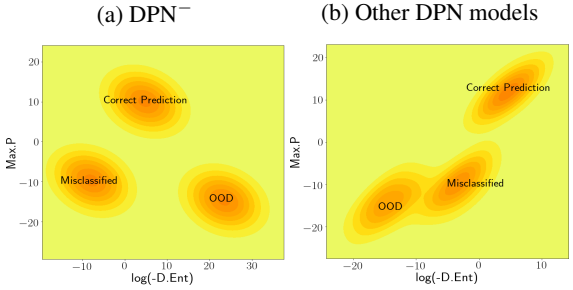


Table 3: KL-divergence scores from the distribution of uncertainty values of missclassified and correctly predicted examples to the OOD examples. See Table 10 (Appendix) for additional results.

OOD	C10				C100				TIM			
	Tiny [29]		LSUN [33]		Tiny [29]		LSUN [33]		CIFAR-10 [28]		CIFAR-100 [28]	
	Miss	Correct	Miss	Correct	Miss	Correct	Miss	Correct	Miss	Correct	Miss	Correct
DPN$_{rev}$	1.4$_{\pm 0.2}$	12.6$_{\pm 0.8}$	2.1$_{\pm 0.3}$	13.9$_{\pm 1.0}$	1.8$_{\pm 0.0}$	6.6$_{\pm 0.1}$	2.6$_{\pm 0.0}$	8.6$_{\pm 0.1}$	9.0$_{\pm 0.6}$	9.2$_{\pm 1.0}$	2.3$_{\pm 0.4}$	3.2$_{\pm 0.5}$
DPN$^+$	1.5$_{\pm 0.2}$	12.1$_{\pm 1.9}$	2.0$_{\pm 0.2}$	12.7$_{\pm 2.0}$	1.8$_{\pm 0.0}$	7.2$_{\pm 0.2}$	2.6$_{\pm 0.0}$	9.1$_{\pm 0.2}$	21.1$_{\pm 3.4}$	27.1$_{\pm 3.9}$	17.7$_{\pm 3.0}$	23.5$_{\pm 3.5}$
DPN$^-$	**2.1**$_{\pm 0.1}$	**20.7**$_{\pm 1.2}$	**2.4**$_{\pm 0.1}$	**22.5**$_{\pm 1.3}$	**44.5**$_{\pm 2.8}$	**244.2**$_{\pm 25.2}$	**49.9**$_{\pm 3.5}$	**272.0**$_{\pm 29.3}$	**729.8**$_{\pm 12.4}$	**1360.4**$_{\pm 94.5}$	**664.9**$_{\pm 11.6}$	**1241.4**$_{\pm 79.8}$

In Table 3, the significantly higher KL-divergence for our DPN$^-$ models indicate that by combining Max.P with D.Ent measure, we can easily distinguish the OOD examples from the in-domain examples. Further, as we consider classification tasks with a larger number of classes, our DPN$^-$ produces more number of fractional concentration parameters for each class for the OOD examples. It further increases the $\log$(-D.Ent) values, leading to maximizing the "gaps" between OOD examples from both in-domain confident predictions as well as misclassified examples.

6 Conclusion

The existing formulation for DPN models often lead to indistinguishable representations between in-domain examples with high data uncertainty among multiple classes and OOD examples. In this work, we have proposed a novel loss function for DPN models that maximizes the representation gap between in-domain and OOD examples. Experiments on benchmark datasets demonstrate that our proposed approach effectively distinguishes the distributional uncertainty from other uncertainty types and outperforms the existing OOD detection models.

7 Broader Impact

Despite the impeccable success of deep neural network (DNN)-based models in various real-world applications, they often produce incorrect predictions without proving any warning for the users. It raises the question of how much can we trust these models and whether it is safe to use them for sensitive real-world applications such as medical diagnosis, self-driving cars or to make financial decisions.

In this paper, we aim to robustly identify the source of uncertainty in the prediction of a DNN based model for classification tasks. Identifying the source of uncertainty in the prediction would allow manual intervention in an informed way and make an AI system more reliable for real-world applications. In particular, we address a shortcoming of the existing techniques and propose a novel solution to improve the detection of anomalous out-of-distribution examples for a classification model.

Acknowledgement

This research is supported by the National Research Foundation, Singapore under its AI Singapore Programme (AISG Award No: AISG-GC-2019-001). Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not reflect the views of National Research Foundation, Singapore

References

[1] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015.

[2] 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.

[3] 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*, 2017.

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

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

[6] JQ Candela, Masashi Sugiyama, Anton Schwaighofer, and Neil D Lawrence. *Dataset shift in machine learning*. The MIT Press, 2009.

[7] Mariusz Bojarski, Davide Del Testa, Daniel Dworakowski, et al. End to end learning for self-driving cars. *arXiv preprint*, 2016.

[8] 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.

[9] Marcos Lopez De Prado. *Advances in financial machine learning*. John Wiley & Sons, 2018.

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

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

[12] Christos Louizos and Max Welling. Multiplicative normalizing flows for variational bayesian neural networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 2218–2227. JMLR. org, 2017.

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

[14] Kimin Lee, Honglak Lee, Kibok Lee, and Jinwoo Shin. Training confidence-calibrated classifiers for detecting out-of-distribution samples. In *ICLR*, 2018.

[15] Dan Hendrycks, Mantas Mazeika, and Thomas Dietterich. Deep anomaly detection with outlier exposure. In *ICLR*, 2019.

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

[17] Max Welling and Yee Whye Teh. Bayesian learning via stochastic gradient langevin dynamics. In *ICML*, 2011.

[18] Murat Sensoy, Lance Kaplan, and Melih Kandemir. Evidential deep learning to quantify classification uncertainty. In *NeurIPS*, 2018.

[19] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv*, 2014.

[20] Shiyu Liang, Yixuan Li, and R. Srikant. Enhancing the reliability of out-of-distribution image detection in neural networks. In *ICLR*, 2018.

[21] Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. In *NeurIPS*, 2018.

[22] Terrance DeVries and Graham W Taylor. Learning confidence for out-of-distribution detection in neural networks. *arXiv*, 2018.

[23] Gabi Shalev, Yossi Adi, and Joseph Keshet. Out-of-distribution detection using multiple semantic label representations. In *NeurIPS*, 2018.

[24] 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 *CVPR*, 2019.

[25] Alexander Meinke and Matthias Hein. Towards neural networks that provably know when they don't know. In *International Conference on Learning Representations*, 2020.

[26] Julian Bitterwolf, Alexander Meinke, and Matthias Hein. Provable worst case guarantees for the detection of out-of-distribution data. *arXiv preprint*, 2020.

[27] Andrey Malinin. Uncertainty estimation in deep learning with application to spoken language assessment. In *Doctoral thesis*, 2019.

[28] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. *Citeseer*, 2009.

[29] Fei-Fei Li, Andrej Karpathy, and Justin Johnson. Tiny imagenet visual recognition challenge. *Stanford University CS231N*, 2017.

[30] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition CVPR*, 2009.

[31] Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. *ICLR*, 2017.

[32] Adam Coates, Andrew Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In *AISTATS*, 2011.

[33] 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*, 2015.

[34] 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.

[35] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. 2017.

[36] John A Swets. *Signal detection theory and ROC analysis in psychology and diagnostics: Collected papers*. Psychology Press, 2014.

[37] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[38] 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.

[39] Khanh Nguyen and Brendan O'Connor. Posterior calibration and exploratory analysis for natural language processing models. 2015.

Appendix

Organization: We organize the appendix as follows:

1. We present a set of ablation studies for our models in Section A.
2. Section B.1 provides the implementation details for our experiments on synthetic datasets.

   Section B.2 provides the experimental setup, implementation details of our models, and competitive models along with the description of the OOD test datasets for our experiments on the benchmark image classification datasets.

   Section B.3 presents a comparative study for confidence calibration performance of different models.
3. The expressions for differential entropy, mutual information of a Dirichlet distribution, and the KL Divergence between two Gaussian distributions are provided in Section C.
4. The extended results (mean $\pm$ standard deviation of 3 models) for the benchmark image classification datasets are provided in Section D from Table 9 to Table 13.

A Ablation Studies

A.1 Different choices for λ_{in} and λ_{out}

Choosing both λ_{in} and λ_{out} to 0 lead Eqn 12 to the same loss function as non-Bayesian outlier exposure (OE) framework [15], while loosing control over the precision of the output Dirichlet distributions. In contrast, setting either λ_{in} or λ_{out} to 0 loses control over the precision for in-domain or OOD examples respectively. We train two additional DPN models, denoted as DPN$_{\{0,-0.5\}}$ and DPN$_{\{0.5,0\}}$ for C100 classification tasks to investigate the choices of these hyper-parameters. DPN$_{\{0,-0.5\}}$ is trained using $\lambda_{in} = 0$ and $\lambda_{out} = -0.5$. DPN$_{\{0.5,0\}}$ is trained using $\lambda_{in} = 0.5$ and $\lambda_{out} = 0$. In Table 4, we present their comparative performance with DPN$^-$.

Table 4: AUROC scores for OOD image detection of our DPN$^-$ models using different values of λ_{in} and λ_{out} for C100 classification task. We report (mean $\pm$ standard deviation) values of three runs.

OOD	Tiny				STL-10				LSUN			
	Max.P	MI	α_0	D.Ent	Max.P	MI	α_0	D.Ent	Max.P	MI	α_0	D.Ent
DPN$_{\{0,-0.5\}}$	84.6$_{\pm 0.0}$	91.1$_{\pm 0.1}$	91.6$_{\pm 0.2}$	21.7$_{\pm 0.7}$	90.2$_{\pm 0.1}$	95.0$_{\pm 0.0}$	95.7$_{\pm 0.1}$	12.1$_{\pm 0.3}$	88.5$_{\pm 0.2}$	93.8$_{\pm 0.1}$	94.2$_{\pm 0.1}$	17.5$_{\pm 0.3}$
DPN$_{\{0.5,0\}}$	89.2$_{\pm 0.4}$	93.6$_{\pm 0.1}$	93.6$_{\pm 0.1}$	93.6$_{\pm 0.1}$	92.1$_{\pm 0.5}$	96.2$_{\pm 0.3}$	96.2$_{\pm 0.3}$	96.0$_{\pm 0.2}$	92.0$_{\pm 0.3}$	95.9$_{\pm 0.1}$	95.9$_{\pm 0.1}$	95.9$_{\pm 0.1}$
DPN$^-$	89.2$_{\pm 0.1}$	**94.5$_{\pm 0.1}$**	**94.5$_{\pm 0.1}$**	38.1$_{\pm 0.5}$	92.8$_{\pm 0.1}$	**96.8$_{\pm 0.1}$**	**96.8$_{\pm 0.1}$**	25.4$_{\pm 0.4}$	92.8$_{\pm 0.1}$	**96.5$_{\pm 0.1}$**	**96.5$_{\pm 0.1}$**	31.5$_{\pm 0.4}$

Analyzing DPN$_{\{0,-0.5\}}$: The choice of only $\lambda_{in} = 0$ in Eqn. 12 does not enforce the DPN to produce larger concentration parameters for the in-domain examples. It only learns to produce fractional (i.e <1) concentration parameters for OOD examples, leading to produce sharper multi-modal Dirichlet distributions for OOD examples.

However, now the network can produce fractional (i.e <1) concentration parameters even for in-domain examples as well. This leads to inappropriately interpolate the concentration parameters in the boundary of in-domain and OOD regions. As a result, it leads to degrading the OOD detection performance. We can see that, similar to DPN$^-$ models, DPN$_{\{0,-0.5\}}$ models also produce lower AUROC scores for D.Ent. This indicates that the choice of $\lambda_{in} = 0, \lambda_{out} < 0$ leads to produce sharp multi-modal Dirichlet distributions for OOD examples. However, their overall OOD detection performance degrade compare to DPN$^-$ models.

Analyzing DPN$_{\{0.5,0\}}$: On the other hand, DPN$_{\{0.5,0\}}$ demonstrates similar property as DPN$^+$. In this case, the network produces flatter Dirichlet distributions for OOD examples compare to the in-domain examples. As we can see that DPN$_{\{0.5,0\}}$ produces high AUROC scores for D.Ent measure. However, as before, it does not address the issue of efficiently maximizing the *'representational gap'* between in-domain and OOD examples. We can see in Table 4, DPN$_{\{0.5,0\}}$ cannot exceed the OOD detection performance of DPN$^-$ models, similar to the DPN$^+$ models.

A.2 A different choice of β_{out} for RKL loss

In section 4, we explain that choosing fractional values for target concentration parameters, $\boldsymbol{\beta}_{out}$ for RKL loss [16] does not guarantee to produce fractional concentration parameters for OOD examples (see Eq. 9). Here, we investigate this by choosing the target concentration parameters to 0.1 for all classes for OOD training examples. For in-domain training examples, we set the target concentration parameters as 100 for the correct class and 1 for the incorrect classes. We denote it as DPN$^{0.1}_{rev}$.

In Table 5, we compare their OOD detection performance with the standard DPN$_{rev}$ models where the target concentration parameters for OOD examples are set to 1 for all classes. We observe that the performance of DPN$^{0.1}_{rev}$ models produce lower AUROC scores for D.Ent measures, while their overall performance degrade compare to the standard DPN$_{rev}$ models. This is because DPN$^{0.1}_{rev}$ models often produce both greater than and less than 1 values of concentration parameters of OOD examples, that leads to uni-modal Dirichlet distributions, instead of a multi-modal Dirichlet. This representation is often similar to the in-domain examples. Hence, it becomes even more difficult to distinguish the in-domain and OOD examples for DPN$^{0.1}_{rev}$ models, which lead to degrade their overall performance.

Table 5: AUROC scores OOD image detection results for DPN models using RKL loss function [16] with different choices of hyper-parameters for C100 classification task. We report (mean $\pm$ standard deviation) values of three runs.

OOD	Tiny				STL-10				LSUN			
	Max.P	MI	α_0	D.Ent	Max.P	MI	α_0	D.Ent	Max.P	MI	α_0	D.Ent
DPN$^{0.1}_{rev}$	74.9$_{\pm 0.2}$	80.4$_{\pm 0.2}$	80.7$_{\pm 0.2}$	48.1$_{\pm 0.1}$	78.1$_{\pm 0.1}$	84.3$_{\pm 0.1}$	84.7$_{\pm 0.1}$	32.8$_{\pm 0.2}$	76.7$_{\pm 0.1}$	82.2$_{\pm 0.0}$	82.5$_{\pm 0.1}$	49.3$_{\pm 0.2}$
DPN$_{rev}$	81.2$_{\pm 0.2}$	**83.8$_{\pm 0.1}$**	**83.8$_{\pm 0.1}$**	83.5$_{\pm 0.1}$	87.2$_{\pm 0.1}$	**89.3$_{\pm 0.1}$**	**89.3$_{\pm 0.1}$**	89.0$_{\pm 0.1}$	86.7$_{\pm 0.0}$	**89.3$_{\pm 0.1}$**	**89.3$_{\pm 0.1}$**	88.9$_{\pm 0.1}$

A.3 A Binary Classifier for OOD Detection

In this work, we show that in the presence of high data uncertainty, the existing OOD detectors often lead to the same representation for in-domain examples as the OOD examples. Hence, one can simply think of training a binary classifier using in-domain and OOD training examples as two different classes to distinguish between in-domain examples and OOD examples. Since it does not need to classify the in-domain examples among multiple classes, it would not suffer from data uncertainty and should automatically solve the problem. However, note that such a binary classifier only learns to produce sharp categorical distributions for these training examples. Hence, given an unknown OOD test example, it does not necessarily produce sharp categorical distribution for the OOD class.

For example, for our experiment on C10 classification task, we use CIFAR-10 training images as the in-domain training set and CIFAR-100 training examples the OOD training set. In contrast, for C100 classification task, we use CIFAR-100 training images as the in-domain training set and CIFAR-10 training examples as the OOD training set. Hence, given an OOD test example from TIM dataset, if the binary classifier for C10 produces a higher probability score for the OOD class, it is expected to produce a lower probability score for the OOD class for C100 classification task. In contrast, our DPN$^-$ models are explicitly trained to produce multi-modal Dirichlet distributions for an unknown test example, whenever it does not 'fit' into the in-domain class-labels.

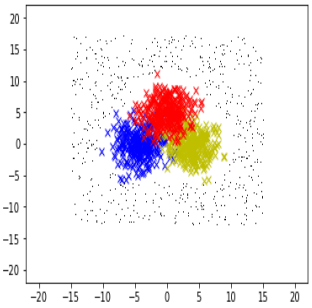
Table 6: OOD image detection results of the binary classifiers compare to our DPN$^-$ models for C10 and C100 classification task. We report (mean $\pm$ standard deviation) values of three runs.

OOD	C-10 Classification				C-100 classification			
	AUROC		AUPR		AUROC		AUPR	
	Binary	DPN$^-$	Binary	DPN$^-$	Binary	DPN$^-$	Binary	DPN$^-$
Tiny	**99.0$_{\pm 0.2}$**	**99.0$_{\pm 0.1}$**	**99.1$_{\pm 0.1}$**	**99.1$_{\pm 0.1}$**	84.8$_{\pm 1.2}$	**94.5$_{\pm 0.1}$**	87.9$_{\pm 0.7}$	**95.1$_{\pm 0.1}$**
STL-10	**93.7$_{\pm 0.9}$**	85.3$_{\pm 0.5}$	**94.0$_{\pm 0.7}$**	85.2$_{\pm 0.6}$	90.7$_{\pm 0.9}$	**96.8$_{\pm 0.1}$**	91.0$_{\pm 0.6}$	**96.7$_{\pm 0.1}$**
LSUN	98.9$_{\pm 0.2}$	**99.3$_{\pm 0.0}$**	98.9$_{\pm 0.2}$	**99.3$_{\pm 0.1}$**	91.2$_{\pm 1.2}$	**96.5$_{\pm 0.1}$**	93.0$_{\pm 0.8}$	**97.0$_{\pm 0.1}$**
Places365	98.8$_{\pm 0.2}$	**98.9$_{\pm 0.1}$**	**99.7$_{\pm 0.0}$**	**99.7$_{\pm 0.0}$**	88.2$_{\pm 1.3}$	**94.5$_{\pm 0.1}$**	96.7$_{\pm 0.3}$	**98.4$_{\pm 0.0}$**
Textures	**99.7$_{\pm 0.1}$**	**99.7$_{\pm 0.0}$**	**99.6$_{\pm 0.1}$**	99.4$_{\pm 0.1}$	65.4$_{\pm 1.5}$	**85.2$_{\pm 0.1}$**	65.2$_{\pm 0.9}$	**78.9$_{\pm 0.2}$**

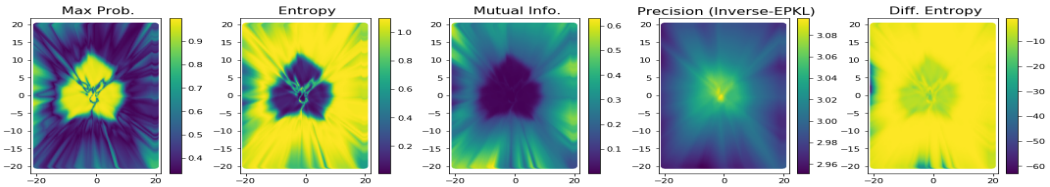
In Table 6, we compare the OOD detection performance of such binary classifiers with our DPN$^-$ models for C10 and C100. For the binary classifier, we consider the probability score of the "in-domain" class as their uncertainty metric. For our DPN$^-$ models, we select the best AUROC and

AUPR values from Table 11 and Table 12 for C10 and C100 classification tasks respectively.. We can see that, while the binary classifier often out-performs our DPN$^-$ models, it does not necessarily provide the upper bound for the OOD detection tasks. In practice, since we do not know the characteristics of the OOD test examples, it may not be suitable to use a binary classifier for OOD detection tasks.

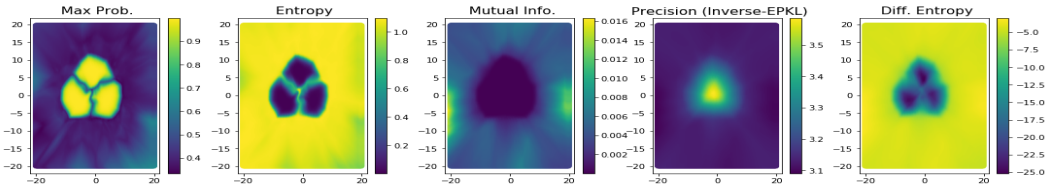
Figure 7: Visualization and understanding the desired characteristics of different DPN models. We visualize the uncertainty measures for different data-points for DPN$_{rev}$, DPN$^+$ and DPN$^-$.



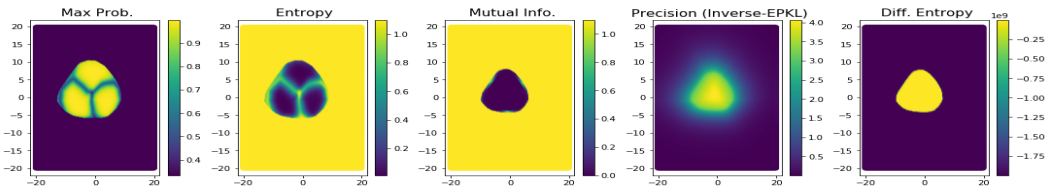
(a) In-domain & OOD training examples



(b) DPN$_{rev}$



(c) DPN$^+$



(d) DPN$^-$

B Implementation Details and Extended Results

B.1 Synthetic Datasets

The three classes of our synthetic dataset are 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 sample 200 training data points from each distribution for each class. We also draw 600 samples for OOD training examples from a uniform distribution of $\mathcal{U}([-15, 15], [-13, 17])$, outside the Gaussian distributions.

Hyper-parameters. We train a neural network with 2 hidden layers with 125 nodes each and $relu$ activation function. The training code is provided along with the supplementary materials.

We use a neural network with 2 hidden layers with 50 nodes each and ReLU activation function. We set $\gamma = 1.0$ in the overall loss function. We have two DPN models. Our first DPN model, DPN$^+$, is trained using a positive $\lambda_{out} = \frac{1}{\#class} + 0.5$ and $\lambda_{in} = 1.5$. The second DPN model, DPN$^-$, is trained using a negative $\lambda_{out} = \frac{1}{\#class} - 0.5$, and $\lambda_{in} = 0.5$. Our DPN$^+$ and DPN$^-$ models are trained using the SGD optimizer.

We also train a DPN$_{rev}$ model by using RKL loss [16]. We set the concentration hyper-parameters as follows: for in-domain training examples, we set the concentration parameters to $1e2$ for the correct class and 1 for the incorrect classes. For the OOD training examples, we set the concentration parameters as 1 for all classes. We train the DPN$_{rev}$ model using ADAM optimizer [37]. Notably, we could not train the DPN$_{rev}$ model using SGD optimization due to their complex RKL loss function.

Additional Results. We visualize the uncertainty measures of different data points for DPN$_{rev}$ along with our DPN$^+$ and DPN$^-$ models in Figure 7(b), 7(c) and 7(d) respectively. We observe very similar characteristics for DPN$_{rev}$ and our DPN$^+$. In contrast, our DPN$^-$ produces much sharper boundaries to distinguish the in-domain and OOD examples using distributional uncertainty measure i.e mutual information and precision (or inverse-EPKL) to demonstrate its superiority.

We also present the results for $entropy$ of categorical posterior distributions, $\mathcal{H}[p(\omega_c|\boldsymbol{x}^*, D)]$, for different data points. This is a total uncertainty measure as it is derived from the expected predictive categorical distribution, $p(\omega_c|\boldsymbol{x}^*, D)$ i.e by marginalizing $\boldsymbol{\mu}$ and $\boldsymbol{\theta}$ in Eq. 2.

B.2 Benchmark Image Classification Datasets

We use the VGG-16 network for C10, C100, and TIM classification tasks. For C10, we use CIFAR-10 training images ($50,000$ images) as our in-domain training data and CIFAR-100 training images ($50,000$ images) as our OOD training data.

For C-100, we use CIFAR-100 training images ($50,000$ images) as our in-domain training data and CIFAR-10 training images ($50,000$ images) as our OOD training data.

For TIM, we use TIM training images (100,000 images) as our in-domain training data and ImageNet-25K images ($25,000$ images) as our OOD training data. ImageNet-25K is obtained by randomly selecting $25,000$ images from the ImageNet dataset [30].

Hyper-parameters for our DPN$^+$ & DPN$^-$ models. Similar to [5, 16, 15], we do not need to tune any hyper-parameters during testing. In other words, the OOD test examples remain unknown to our DPN classifiers, as in a real-world scenario. We set $\gamma = 0.5$ for our loss function in Eqn. 12, as applied in [15].

We train two different DPN models for each classification task, using both positive and negative values for λ_{out} to analyze the effect of flat Dirichlet distributions and sharp Dirichlet distributions across the edges of the simplex respectively for the OOD examples. Note that we cannot choose arbitrarily large values for λ_{in} and λ_{out} as it would degrade the in-domain classification accuracy of the models. For our experiments in the main paper, we select the hyper-parameters as follows: Our first model, DPN$^+$ is trained with positive $\lambda_{out} = \frac{1}{\#class} + 0.5$ and $\lambda_{in} = 1.5$. The second model, DPN$^-$ is trained with negative $\lambda_{out} = \frac{1}{\#class} - 0.5$, and $\lambda_{in} = 0.5$.

Competitive Systems: Implementations and Additional Discussions. We compare the performance of our models with standard DNN as baseline model [31], Bayesian Monte-Carlo dropout (MCDP) [11], deep-ensemble models (DE) [13] and evidential deep learning (EDL) [18], DPN$_{fwd}$ and DPN$_{rev}$ using the forward and reverse KL-divergence loss function proposed in [5] and [16] respectively, non-Bayesian frameworks such as outlier exposure (OE) [15]. We use the same architecture as our DPN models for the other competitive models.

MCDP models: For MCDP models, we use the standard DNN (baseline) model with randomly dropping the nodes during test time. The predictive categorical distributions are obtained by averaging the outputs for 10 iterations. For OE, DPN$_{fwd}$ and DPN$_{rev}$ models, we use the same setup as applied

Table 7: Details of Train and Test Datasets used for the different classifiers.

Classifier	Input	#Classes	Training Datasets		Test Datasets	
			In-Domain	OOD	In-Domain	OOD
C10	32×32	10	CIFAR-10 Training Set (50,000 images)	CIFAR-100 Training Set (50,000 images)	CIFAR-10 Test Images (10,000 Images)	Tiny, STL-10, LSUN etc.
C100	32×32	100	CIFAR-100 Training Set (50,000 images)	CIFAR-10 Training Set (50,000 images)	CIFAR-100 Test Set (10,000 Images)	Tiny, STL-10, LSUN etc.
TIM	64×64	200	TinyImageNet Training Set (100,000 images)	ImageNet-25K (25,000 randomly sampled images from ImageNet)	TIM Test Images (10,000 Images)	CIFAR-10, CIFAR-100, Textures etc.

for our DPN models (See Table 7). Table 9 presents the classification accuracies achieved by different models for different classification tasks, along with their performance for misclassified example detection. For DE, we use an ensemble of three baseline DNN models for our experiments.

EDL models: Similar to DPN models, EDL also produces Dirichlet distribution for the input examples [18]. Unlike DPN, EDL applies ReLU activation, instead of the exponential function (Eqn. 5), to induce non-negative constraints to produce the concentration parameters of their output Dirichlet distributions.

EDL models are trained using the in-domain examples. Their network is trained using a loss function that explicitly maximizes the concentration parameter of the correct class while minimizing the overall precision of the Dirichlet for each in-domain training examples. For C10 classifiers, we train the EDL models from the scratch. For C100 and TIM classifiers, we initialize the EDL models using the pre-trained Baseline models to achieve competitive in-domain classification accuracy. Next, we replace the soft-max using ReLU activation and retrain the models using the proposed loss function for EDL.

Since both DPN and EDL models output Dirichlet distribution for an input example, we can define the same uncertainty measures as the DPN models. In Table 9 and Table 11-13, we present the results of the EDL models. Interestingly, in Table 9, we observe that EDL models often tend to produce lower AUROC and AUPR scores under distributional uncertainty measures for the misclassification detection task. This property is desirable (see Section 4 and 5.1). However, they achieve significantly lower OOD detection performance compared to the state-of-the-art competitive models (Table 11-13). Further, in Table 8, we observe that the calibration performance of EDL models are also dropped than the other state-of-the-art OOD detection models.

Existing DPN models [5, 16]: DPN$_{fwd}$ and DPN$_{rev}$ models are trained only using the ADAM optimizer for all classification tasks [37]. We could not use the SGD optimizer to train these models due to the complex RKL loss. In contrast, we have not encountered such a problem for other models.

For example, we use SGD to train the other models for C10 classification task. We observe that both DPN$_{fwd}$ and DPN$_{rev}$ models achieve lower classification accuracy than the other classifiers for this task (Table 9). For C100 and TIM classification tasks, we choose the ADAM optimizer for all models. We find that all the OOD detection models achieve similar classification accuracy for these two tasks (Table 9).

Further, note that, the choice of larger values for the hyper-parameter, β for the RKL in Eqn. 8 makes it difficult to optimize the network. For our experiments, we choose the same set of hyper-parameters for DPN$_{fwd}$ and DPN$_{rev}$ models as suggested in the original paper [16]. The concentration parameters for in-domain training examples are set to 100 for the correct class and 1 for the incorrect classes. For OOD training examples, we choose the concentration parameters as 1 for all classes.

B.2.1 Description of the OOD Test Datasets

We use a wide-range of OOD datasets for our experiments, as described in the following. For C10 and C100 classifiers, these input test images are resized to 32×32, while for TIM classifiers, we resize them to 64×64.

TinyImageNet (Tiny) [29]. This dataset is used as an OOD test dataset *only* for C-10 and C-100 classifiers. Note that, for TinyImageNet classifiers, this is the in-domain test set.

This is a subset of the Imagenet dataset. We use the validation set, that contains $10,000$ test images from 200 different image classes for our evaluation during test time.

CIFAR-10 and CIFAR-100 [28]. This dataset is used as the OOD test dataset *only* for TIM classifiers. We use the validation set, that contains $10,000$ test images from 10 and 100 image classes respectively.

LSUN [33]. The Large-scale Scene UNderstanding dataset (LSUN) contains images of 10 different scene categories. We use its validation set, containing $10,000$ images, as an unknown OOD test set.

Places 365 [38]. The validation set of this dataset consists of 36500 images of 365 scene categories.

Textures [34] contains 5640 textural images in the wild belonging to 47 categories.

STL-10 contains $8,000$ images of natural images from 10 different classes [32].

B.3 Results for Confidence Calibration

Calibration error measures if the confidence estimates produced by the classifier for its predictions misrepresent the empirical performance [39, 35, 15]. A well-calibrated classifier should produce the confidence probabilities that matches with the empirical frequency of correctness. For example, if a classifier predicts an event with 90% probability, we would like it to be corrected for 90% of the time. However, several studies have demonstrated that DNN classifiers tend to produce over-confidence in their predictions.

Several measures have been proposed to compute the calibration error of a classifier [35, 15]. In this paper, we use the Root Mean Square (RMS) Calibration Error that computes the square root of the expected squared difference between confidence and accuracy at a confidence level [15]. Since the confidence values can be distributed non-uniformly, Hendrycks et al. [15] proposed to partition the samples into multiple bins with dynamic ranges and measure the average confidence and accuracy of each bin to compute the calibration error.

Table 8: Root mean square (RMS) calibration error. Lower scores are better.

	C10	C100	TIM
Baseline	16.2±0.0	6.6±0.3	5.2±0.0
MCDP	15.7±0.1	6.7±0.0	5.3±0.2
DE	16.1±NA	6.8±NA	6.2±NA
EDL	14.9±0.2	17.7±0.6	12.4±0.1
OE	6.4±0.4	3.8±0.1	4.2±0.1
DPN_{rev}	9.2±0.4	10.4±0.1	7.2±0.5
DPN^{+}	**6.3**±**0.3**	4.3±0.0	2.8±0.3
DPN^{-}	6.5±0.2	**3.5**±**0.1**	**2.7**±**0.3**

A real-world classifier should provide calibrated probabilities on both in- and out-of-distribution examples. Hence, Hendrycks et al. [15] proposed to incorporate OOD test examples in the calculation of the RMS calibration error. Since the OOD examples do not belong to any of the in-domain classes, these examples are always considered to be incorrectly classified. Hence, the classifier should produce low confidence for these OOD test examples.

In Table 8 we present a comparative results for RMS calibration error [15]. We take an equal number of OOD test examples as the in-domain test samples for this experiment. For C10 and C100 classifiers, we use $5,000$ in-domain test examples from and take $5,000$ OOD test examples from STL-10 dataset. For TIM classifiers, we use $5,000$ in-domain test examples and $5,000$ OOD test examples from CIFAR-100. We apply the soft-max temperature scaling to report the calibration error in Table 8. Note that, EDL models do not apply soft-max activation to produce their probability scores. Hence, we use 'temperature translating' where we instead add the temperature parameter to the logit outputs. We can Table 8 that our proposed DPN^{-} models achieve comparable performance with the OE and DPN^{+} models for C10. For C100 and TIM, our DPN^{-} models outperform other comparative systems.

C Derivations of different measures

C.1 Differential Entropy for a Dirichlet

Differential Entropy of a Dirichlet distribution can be calculated as follows:

$$\begin{aligned}\mathcal{H}[p(\boldsymbol{\mu}|\boldsymbol{x}^*, D_{in})] &= -\int p(\boldsymbol{\mu}|\boldsymbol{x}^*, D_{in}) \ln p(\boldsymbol{\mu}|\boldsymbol{x}^*, D_{in}) d\boldsymbol{\mu} \\ &= \sum_{c=1}^{K} \ln \Gamma(\alpha_c) - \ln \Gamma(\alpha_0) - \sum_{c=1}^{K} (\alpha_c - 1)(\psi(\alpha_c) - \psi(\alpha_0))\end{aligned} \tag{14}$$

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

C.2 Mutual Information of a Dirichlet

The mutual information of the labels y and the categorical $\boldsymbol{\mu}$ of a DPN is computed as:

$$\mathcal{I}[y, \boldsymbol{\mu}|\boldsymbol{x}^*, \hat{\boldsymbol{\theta}}] = \sum_{c=1}^{K} \frac{\alpha_c}{\alpha_0} \big[\psi(\alpha_c + 1) - \psi(\alpha_0 + 1) - \ln \frac{\alpha_c}{\alpha_0}\big] \tag{15}$$

C.3 KL Divergence between two Gaussians

The KL divergence from a Gaussian distribution $\mathcal{N}_1(\boldsymbol{\mu}_1, \Sigma_1)$ to Gaussian distribution, $\mathcal{N}_2(\boldsymbol{\mu}_2, \Sigma_2)$ is computed as follows:

$$KL(\mathcal{N}_2 \;||\; \mathcal{N}_1) := \frac{1}{2}\Big[tr(\Sigma_1^{-1}\Sigma_2) \;\; - d + \frac{det(\Sigma_1)}{det(\Sigma_0)} + (\boldsymbol{\mu_1} - \boldsymbol{\mu_2})^T \Sigma_1^{-1}(\boldsymbol{\mu_1} - \boldsymbol{\mu_2})\Big] \tag{16}$$

where, d is the dimension of $\boldsymbol{\mu}_1$ or $\boldsymbol{\mu}_2$. $det(\Sigma)$ represents the determinant of Σ. tr computes the trace of a matrix.

D Extended Results

In the following, we present an extended version of the results for a wide range of OOD test datasets along with an additional uncertainty measure i.e, entropy. We also report the results for the deep ensemble (DE) framework using an ensemble of 3 models.

In Table 9, we present the results for misclassification detection along with the classification accuracy for different approaches. Note that the DPN models achieve higher AUROC and AUPR scores even for distributional uncertainty measures such as MI, precision (α_0), and D.Ent.

In Table 10, we present an extended version of Table 3 along with additional OOD datasets for each classification tasks. In Table 11-13, we present the additional results for OOD detection performance.

Table 9: Classification accuracy and misclassified image detection. Here, we report the (mean $\pm$ standard deviation) of 3 runs for each framework. Note that AUPR may *not* be an ideal metric for comparison, as it depends on the *base rates* i.e the number of misclassified examples v.s correctly classified predictions. That is, AUPR scores are comparable when the models achieve similar classification accuracy. Our DPN$^-$ models achieve comparable performance foe misclassified image detection using the AUROC metric.

(a) C10 classification task

	AUROC					AUPR					Acc.
	Max.P	Ent	MI	α_0	D.Ent	Max.P	Ent	MI	α_0	D.Ent	
Baseline	93.3$_{\pm 0.1}$	93.4$_{\pm 0.1}$	-	-	-	43.9$_{\pm 0.7}$	47.0$_{\pm 0.3}$	-	-	-	94.1$_{\pm 0.0}$
MCDP	**93.6$_{\pm 0.2}$**	**93.6$_{\pm 0.2}$**	93.2$_{\pm 0.1}$	-	-	46.1$_{\pm 2.0}$	46.5$_{\pm 1.9}$	40.9$_{\pm 1.5}$	-	-	94.2$_{\pm 0.1}$
DE	93.5$_{\pm NA}$	93.5$_{\pm NA}$	92.7$_{\pm NA}$	-	-	45.6$_{\pm NA}$	46.6$_{\pm NA}$	39.8$_{\pm NA}$	-	-	94.0 $_{\pm NA}$
EDL	91.3$_{\pm 0.0}$	91.2$_{\pm 0.0}$	80.1$_{\pm 0.5}$	76.7$_{\pm 0.7}$	89.7$_{\pm 0.0}$	44.8$_{\pm 0.2}$	43.8$_{\pm 0.1}$	21.7$_{\pm 0.5}$	18.1$_{\pm 0.5}$	37.9$_{\pm 0.1}$	93.1$_{\pm 0.0}$
OE	92.0$_{\pm 0.0}$	91.6$_{\pm 0.0}$	-	-	-	35.3$_{\pm 0.8}$	33.6$_{\pm 0.8}$	-	-	-	94.2$_{\pm 0.1}$
DPN$_{fwd}$	90.3$_{\pm 0.2}$	90.1$_{\pm 0.2}$	88.6$_{\pm 0.2}$	88.0$_{\pm 0.2}$	88.0$_{\pm 0.1}$	49.2$_{\pm 0.9}$	47.7$_{\pm 0.8}$	43.3$_{\pm 0.4}$	41.6$_{\pm 0.4}$	40.8$_{\pm 0.6}$	88.3$_{\pm 0.2}$
DPN$_{rev}$	89.6$_{\pm 0.1}$	89.4$_{\pm 0.1}$	88.7$_{\pm 0.2}$	88.7$_{\pm 0.2}$	89.0$_{\pm 0.2}$	**50.0$_{\pm 0.8}$**	48.8$_{\pm 0.7}$	46.1$_{\pm 0.9}$	45.8$_{\pm 0.8}$	47.7$_{\pm 0.7}$	90.6$_{\pm 0.0}$
DPN$^+$	92.2$_{\pm 0.3}$	91.7$_{\pm 0.3}$	90.3$_{\pm 0.1}$	90.3$_{\pm 0.1}$	90.5$_{\pm 0.2}$	36.6$_{\pm 0.5}$	34.9$_{\pm 0.7}$	31.2$_{\pm 0.8}$	31.2$_{\pm 0.8}$	31.6$_{\pm 0.7}$	94.0$_{\pm 0.1}$
DPN$^-$	92.6$_{\pm 0.1}$	92.2$_{\pm 0.1}$	89.9$_{\pm 0.0}$	89.9$_{\pm 0.0}$	66.2$_{\pm 0.7}$	37.2$_{\pm 0.7}$	35.1$_{\pm 0.6}$	31.3$_{\pm 0.4}$	30.6$_{\pm 0.4}$	17.1$_{\pm 0.4}$	**94.4$_{\pm 0.0}$**

(b) C100 classification task

	AUROC					AUPR					Acc.
	Max.P	Ent	MI	α_0	D.Ent	Max.P	Ent	MI	α_0	D.Ent	
Baseline	86.8$_{\pm 0.1}$	87.0$_{\pm 0.1}$	-	-	-	68.4$_{\pm 0.4}$	69.2$_{\pm 0.3}$	-	-	-	72.3$_{\pm 0.0}$
MCDP	87.2$_{\pm 0.0}$	**87.3$_{\pm 0.0}$**	83.3$_{\pm 0.3}$	-	-	69.1$_{\pm 0.3}$	69.3$_{\pm 0.3}$	53.9$_{\pm 0.5}$	-	-	**72.7$_{\pm 0.1}$**
DE	87.0$_{\pm NA}$	87.1$_{\pm NA}$	83.4$_{\pm NA}$	-	-	69.2$_{\pm NA}$	**69.7$_{\pm NA}$**	56.2$_{\pm NA}$	-	-	72.2$_{\pm NA}$
EDL	85.8$_{\pm 0.3}$	85.0$_{\pm 0.3}$	44.4$_{\pm 1.0}$	43.4$_{\pm 1.1}$	55.7$_{\pm 0.7}$	69.3$_{\pm 1.1}$	68.5$_{\pm 1.0}$	28.5$_{\pm 1.0}$	28.0$_{\pm 1.0}$	36.3$_{\pm 1.4}$	70.4$_{\pm 0.3}$
OE	86.9$_{\pm 0.0}$	86.9$_{\pm 0.1}$	-	-	-	67.7$_{\pm 0.3}$	66.9$_{\pm 0.4}$	-	-	-	71.6$_{\pm 0.0}$
DPN$_{rev}$	79.3$_{\pm 0.1}$	78.5$_{\pm 0.1}$	73.5$_{\pm 0.1}$	73.1$_{\pm 0.1}$	75.7$_{\pm 0.1}$	65.3$_{\pm 0.4}$	64.1$_{\pm 0.3}$	58.4$_{\pm 0.3}$	57.9$_{\pm 0.3}$	61.2$_{\pm 0.3}$	71.1$_{\pm 0.1}$
DPN$^+$	86.5$_{\pm 0.1}$	86.5$_{\pm 0.1}$	81.2$_{\pm 0.0}$	81.3$_{\pm 0.0}$	81.9$_{\pm 0.1}$	66.8$_{\pm 0.3}$	66.3$_{\pm 0.3}$	57.8$_{\pm 0.2}$	57.8$_{\pm 0.2}$	59.2$_{\pm 0.3}$	72.1$_{\pm 0.1}$
DPN$^-$	86.4$_{\pm 0.1}$	86.5$_{\pm 0.1}$	82.3$_{\pm 0.0}$	82.3$_{\pm 0.0}$	81.7$_{\pm 0.1}$	67.0$_{\pm 0.5}$	66.6$_{\pm 0.3}$	58.9$_{\pm 0.2}$	58.9$_{\pm 0.2}$	59.1$_{\pm 0.2}$	72.3$_{\pm 0.1}$

(c) TIM classification task

	AUROC					AUPR					Acc.
	Max.P	Ent	MI	α_0	D.Ent	Max.P	Ent	MI	α_0	D.Ent	
Baseline	86.7$_{\pm 0.0}$	86.8$_{\pm 0.1}$	-	-	-	77.2$_{\pm 0.1}$	77.1$_{\pm 0.3}$	-	-	-	62.5$_{\pm 0.2}$
MCDP	86.6$_{\pm 0.1}$	86.4$_{\pm 0.1}$	83.3$_{\pm 0.3}$	-	-	76.8$_{\pm 0.3}$	76.4$_{\pm 0.3}$	67.2$_{\pm 1.2}$	-	-	**62.7$_{\pm 0.2}$**
DE	**86.8$_{\pm NA}$**	**86.8$_{\pm NA}$**	83.3$_{\pm NA}$	-	-	77.2$_{\pm NA}$	77.0$_{\pm NA}$	67.6$_{\pm NA}$	-	-	62.6$_{\pm NA}$
EDL	85.9$_{\pm 0.2}$	83.6$_{\pm 0.1}$	73.0$_{\pm 0.6}$	72.7$_{\pm 0.6}$	75.5$_{\pm 0.4}$	77.0$_{\pm 0.4}$	73.2$_{\pm 0.4}$	62.1$_{\pm 0.5}$	61.9$_{\pm 0.6}$	64.5$_{\pm 0.4}$	60.9$_{\pm 0.1}$
OE	85.9$_{\pm 0.2}$	85.8$_{\pm 0.1}$	-	-	-	**77.7$_{\pm 0.4}$**	77.3$_{\pm 0.2}$	-	-	-	59.8$_{\pm 0.2}$
DPN$_{rev}$	81.9$_{\pm 0.3}$	81.0$_{\pm 0.2}$	72.2$_{\pm 0.7}$	70.2$_{\pm 0.9}$	78.3$_{\pm 0.3}$	75.0$_{\pm 0.3}$	73.4$_{\pm 0.4}$	61.4$_{\pm 0.9}$	59.2$_{\pm 0.9}$	70.0$_{\pm 0.5}$	60.5$_{\pm 0.2}$
DPN$^+$	85.7$_{\pm 0.2}$	85.7$_{\pm 0.1}$	78.3$_{\pm 0.4}$	78.7$_{\pm 0.5}$	79.7$_{\pm 0.2}$	77.4$_{\pm 0.4}$	76.7$_{\pm 0.2}$	66.3$_{\pm 0.5}$	66.4$_{\pm 0.5}$	68.5$_{\pm 0.4}$	59.7$_{\pm 0.1}$
DPN$^-$	85.4$_{\pm 0.1}$	85.0$_{\pm 0.0}$	79.1$_{\pm 0.5}$	79.4$_{\pm 0.4}$	79.9$_{\pm 0.2}$	76.9$_{\pm 0.1}$	76.2$_{\pm 0.1}$	67.3$_{\pm 0.7}$	67.4$_{\pm 0.7}$	69.4$_{\pm 0.4}$	59.4$_{\pm 0.1}$

Table 10: KL-divergence scores from the distribution of uncertainty values of missclassified and correctly predicted examples to the OOD examples. Higher scores are desirable as it indicates greater gap between in-domain and OOD examples. We report the (mean $\pm$ standard deviation) of 3 runs for each frameworks.

(a) C10 classification task

OOD	STL-10		Tiny		LSUN		Places365		Textures	
	Miss	Correct	Miss	Correct	Miss	Correct	Miss	Correct	Miss	Correct
DPN$_{fwd}$	**1.6**$\pm$**0.1**	1.6$\pm$0.4	1.1$\pm$0.2	6.1$\pm$0.8	1.4$\pm$0.2	6.7$\pm$0.7	1.3$\pm$0.2	6.6$\pm$0.7	2.8$\pm$0.4	12.7$\pm$3.0
DPN$_{rev}$	0.1$\pm$0.0	5.7$\pm$0.7	1.4$\pm$0.2	12.6$\pm$0.8	2.1$\pm$0.3	13.9$\pm$1.0	1.6$\pm$0.2	13.2$\pm$0.8	3.5$\pm$0.1	16.3$\pm$0.6
DPN$^+$	0.3$\pm$0.0	4.7$\pm$0.5	1.5$\pm$0.2	12.1$\pm$1.9	2.0$\pm$0.2	12.7$\pm$2.0	1.5$\pm$0.2	12.2$\pm$1.9	**3.7**$\pm$**0.1**	15.4$\pm$1.9
DPN$^-$	0.5$\pm$0.0	**10.6**$\pm$**0.8**	**2.1**$\pm$**0.1**	**20.7**$\pm$**1.2**	**2.4**$\pm$**0.1**	**22.5**$\pm$**1.3**	**2.1**$\pm$**0.1**	**20.9**$\pm$**1.2**	2.9$\pm$0.1	**20.4**$\pm$**1.2**

(a) C100 classification task

OOD	STL-10		Tiny		LSUN		Places365		Textures	
	Miss	Correct	Miss	Correct	Miss	Correct	Miss	Correct	Miss	Correct
DPN$_{rev}$	2.8$\pm$0.1	9.4$\pm$0.1	1.8$\pm$0.0	6.6$\pm$0.1	2.6$\pm$0.0	8.6$\pm$0.1	2.1$\pm$0.0	7.3$\pm$0.1	1.1$\pm$0.0	4.0$\pm$0.1
DPN$^+$	2.7$\pm$0.0	9.3$\pm$0.3	1.8$\pm$0.0	7.2$\pm$0.2	2.6$\pm$0.0	9.1$\pm$0.2	2.1$\pm$0.0	7.8$\pm$0.2	0.5$\pm$0.0	3.5$\pm$0.1
DPN$^-$	**61.2**$\pm$**3.6**	**330.1**$\pm$**34.2**	**44.5**$\pm$**2.8**	**244.2**$\pm$**25.2**	**49.9**$\pm$**3.5**	**272.0**$\pm$**29.3**	**44.1**$\pm$**3.0**	**241.7**$\pm$**24.8**	**18.1**$\pm$**1.3**	**105.0**$\pm$**10.2**

(a) TIM classification task

OOD	STL-10		CIFAR-10		CIFAR-100	
	Miss	Correct	Miss	Correct	Miss	Correct
DPN$_{rev}$	9.0$\pm$0.6	9.2$\pm$1.0	2.3$\pm$0.4	3.2$\pm$0.5	2.3$\pm$0.4	3.1$\pm$0.5
DPN$^+$	21.1$\pm$3.4	27.1$\pm$3.9	17.7$\pm$3.0	23.5$\pm$3.5	17.3$\pm$3.0	23.1$\pm$3.5
DPN$^-$	**729.8**$\pm$**12.4**	**1360.4**$\pm$**94.5**	**664.9**$\pm$**11.6**	**1241.4**$\pm$**79.8**	**606.9**$\pm$**11.4**	**1134.5**$\pm$**72.1**

OOD	LSUN		Places365		Textures	
	Miss	Correct	Miss	Correct	Miss	Correct
DPN$_{rev}$	9.7$\pm$0.6	9.9$\pm$1.0	8.3$\pm$0.7	8.5$\pm$1.0	4.4$\pm$0.4	4.8$\pm$0.7
DPN$^+$	21.5$\pm$3.4	27.5$\pm$3.9	20.6$\pm$3.3	26.5$\pm$3.8	14.7$\pm$2.6	20.0$\pm$3.0
DPN$^-$	**731.6**$\pm$**12.0**	**1363.4**$\pm$**91.5**	**713.2**$\pm$**11.7**	**1330.5**$\pm$**89.2**	**465.7**$\pm$**8.0**	**873.2**$\pm$**62.7**

Table 11: Results of OOD detection for C10. We report (mean ± standard deviation) of three different models. Description of these OOD datasets are provided in Appendix B.2.1.

	Methods	AUROC					AUPR				
		Max.P	Ent.	MI	α_0	D-Ent	Max.P	Ent.	MI	α_0	D-Ent
Tiny	Baseline	88.9±0.0	89.5±0.0	-	-	-	85.0±0.1	86.7±0.1	-	-	-
	MCDP	88.7±0.1	88.9±0.0	88.1±0.1	-	-	85.2±0.1	85.9±0.1	84.0±0.1	-	-
	DE	88.9±NA	89.0±NA	87.8±NA	-	-	85.0±NA	85.5±NA	83.2±NA	-	-
	EDL	87.6±0.1	88.4±0.2	89.1±0.3	87.6±0.4	89.9±0.2	84.9±0.1	86.6±0.1	88.9±0.3	87.4±0.4	89.0±0.1
	OE	98.2±0.1	98.3±0.1	-	-	-	98.3±0.2	98.3±0.2	-	-	-
	DPN$_{fwd}$	92.8±1.0	93.0±1.0	73.3±0.3	71.3±0.4	93.7±1.0	92.6±1.1	92.9±1.1	59.8±0.7	58.1±0.8	93.4±1.3
	DPN$_{rev}$	97.5±0.5	97.6±0.5	97.8±0.4	97.8±0.4	97.7±0.4	97.5±0.5	97.6±0.4	97.6±0.3	97.6±0.3	97.7±0.4
	DPN$^+$	98.0±0.2	98.0±0.2	98.0±0.2	98.0±0.2	98.0±0.2	98.0±0.3	98.0±0.3	97.9±0.3	97.9±0.3	97.9±0.3
	DPN$^-$	**99.0±0.1**	**99.0±0.1**	**99.0±0.1**	97.7±0.1	6.0±0.3	99.0±0.1	**99.1±0.1**	98.9±0.1	94.9±0.1	32.4±0.1
STL-10	Baseline	75.9±0.0	76.2±0.0	-	-	-	68.5±0.1	69.8±0.1	-	-	-
	MCDP	76.2±0.0	76.2±0.0	76.0±0.0	-	-	69.3±0.0	69.7±0.0	69.4±0.1	-	-
	DE	76.0±NA	76.0±NA	75.6±NA	-	-	68.5±NA	68.9±NA	67.6±NA	-	-
	EDL	72.4±0.1	72.7±0.1	71.7±0.1	70.6±0.1	73.3±0.1	66.8±0.0	68.0±0.0	68.2±0.2	66.7±0.3	69.4±0.0
	OE	81.4±1.2	81.5±1.2	-	-	-	80.8±1.1	80.8±1.0	-	-	-
	DPN$_{fwd}$	71.5±1.3	71.6±1.3	65.7±0.2	64.9±0.2	72.0±1.5	68.2±2.1	68.6±2.1	53.3±0.6	52.5±0.7	68.6±2.7
	DPN$_{rev}$	81.6±1.7	81.7±1.7	82.2±1.7	82.2±1.6	81.9±1.7	81.9±1.7	82.0±1.7	82.5±1.7	82.5±1.6	82.2±1.7
	DPN$^+$	81.6±1.4	81.7±1.3	81.8±1.2	81.8±1.2	81.8±1.2	80.9±1.2	81.0±1.3	81.0±1.2	81.0±1.2	81.0±1.2
	DPN$^-$	84.7±0.4	84.8±0.5	**85.3±0.5**	84.9±0.5	34.6±0.4	84.7±0.6	84.9±0.6	**85.2±0.6**	82.0±0.6	42.4±0.2
LSUN	Baseline	90.3±0.0	91.0±0.0	-	-	-	86.6±0.1	88.5±0.1	-	-	-
	MCDP	90.6±0.0	90.8±0.0	90.2±0.0	-	-	87.5±0.1	88.2±0.0	86.6±0.1	-	-
	DE	90.3±NA	90.4±NA	89.7±NA	-	-	86.5±NA	87.1±NA	85.6±NA	-	-
	EDL	90.3±0.0	91.2±0.0	93.5±0.0	92.6±0.0	93.0±0.0	87.8±0.0	89.7±0.0	93.4±0.0	92.3±0.0	92.4±0.0
	OE	98.4±0.3	98.4±0.3	-	-	-	98.2±0.4	98.2±0.4	-	-	-
	DPN$_{fwd}$	93.5±0.7	93.7±0.8	72.6±0.1	70.6±0.2	94.9±0.8	93.2±0.9	93.5±0.9	58.6±0.3	57.0±0.3	94.4±1.0
	DPN$_{rev}$	98.5±0.4	98.6±0.3	98.7±0.3	98.7±0.3	98.7±0.3	98.3±0.4	98.4±0.3	98.5±0.3	98.5±0.3	98.5±0.3
	DPN$^+$	98.2±0.3	98.3±0.4	98.3±0.4	98.3±0.4	98.3±0.4	98.0±0.5	98.1±0.5	98.0±0.4	98.0±0.4	98.0±0.4
	DPN$^-$	99.2±0.1	99.2±0.1	**99.3±0.0**	98.1±0.1	5.0±0.2	99.1±0.1	**99.2±0.1**	99.1±0.1	95.9±0.0	32.1±0.2
Places365	Baseline	89.4±0.0	90.0±0.0	-	-	-	95.5±0.0	96.1±0.0	-	-	-
	MCDP	89.5±0.0	89.7±0.0	89.0±0.0	-	-	95.7±0.0	96.0±0.0	95.3±0.0	-	-
	DE	89.4±NA	89.5±NA	88.7±NA	-	-	95.5±NA	95.7±NA	95.2±NA	-	-
	EDL	88.5±0.0	89.4±0.0	91.3±0.0	90.3±0.1	91.2±0.0	95.6±0.0	96.2±0.0	97.3±0.0	96.8±0.0	97.1±0.0
	OE	98.1±0.1	98.2±0.2	-	-	-	99.4±0.1	99.4±0.1	-	-	-
	DPN$_{fwd}$	93.3±0.9	93.5±0.8	72.1±0.2	70.1±0.3	94.4±0.9	97.9±0.3	98.0±0.3	82.4±0.6	81.4±0.6	98.2±0.3
	DPN$_{rev}$	97.8±0.4	97.9±0.4	98.0±0.4	98.0±0.4	98.0±0.4	99.3±0.1	99.3±0.1	99.4±0.1	99.4±0.1	99.4±0.1
	DPN$^+$	98.0±0.3	98.0±0.3	98.0±0.3	98.0±0.3	98.0±0.3	99.4±0.1	99.4±0.1	99.4±0.1	99.4±0.1	99.4±0.1
	DPN$^-$	**98.9±0.1**	**98.9±0.1**	**98.9±0.1**	97.7±0.1	6.2±0.4	**99.7±0.0**	**99.7±0.0**	**99.7±0.0**	98.6±0.0	61.0±0.2
Textures	Baseline	88.8±0.0	89.2±0.0	-	-	-	74.9±0.1	76.9±0.2	-	-	-
	MCDP	87.4±0.1	87.5±0.1	85.7±0.2	-	-	73.3±0.2	74.2±0.1	66.9±0.3	-	-
	DE	88.7±NA	88.8±NA	87.2±NA	-	-	74.8±NA	75.4±NA	72.1±NA	-	-
	EDL	84.3±0.4	84.9±0.5	84.9±0.9	83.1±1.0	86.3±0.6	71.5±0.4	73.6±0.5	76.4±1.1	73.9±1.3	77.0±0.6
	OE	99.4±0.0	99.4±0.1	-	-	-	98.9±0.2	98.9±0.3	-	-	-
	DPN$_{fwd}$	98.3±0.4	98.4±0.4	68.2±0.2	65.8±0.1	98.6±0.2	97.1±0.6	97.4±0.6	42.0±0.2	40.4±0.2	97.7±0.6
	DPN$_{rev}$	99.4±0.0	99.3±0.0	99.4±0.0	99.4±0.0	99.4±0.0	98.5±0.0	98.3±0.0	98.5±0.1	98.4±0.1	98.4±0.0
	DPN$^+$	99.5±0.0	99.6±0.0	99.5±0.0	99.5±0.0	99.5±0.0	99.1±0.2	99.1±0.1	99.1±0.1	99.1±0.1	99.1±0.1
	DPN$^-$	**99.7±0.0**	**99.7±0.0**	99.5±0.0	97.8±0.1	3.6±0.1	**99.4±0.0**	**99.4±0.1**	98.8±0.0	90.1±0.3	21.3±0.0

Table 12: Results of OOD image detection for C100. We report (mean ± standard deviation) of three different models. Description of these OOD datasets are provided in Appendix B.2.1.

	Methods	AUROC					AUPR				
		Max.P	Ent.	MI	α_0	D-Ent	Max.P	Ent.	MI	α_0	D-Ent
Tiny	Baseline	68.8±0.2	71.4±0.2	-	-	-	66.6±0.2	70.2±0.2	-	-	-
	MCDP	69.7±0.3	70.2±0.3	70.6±0.3	-	-	67.4±0.3	68.5±0.2	66.0±0.2	-	-
	DE	68.9±NA	69.3±NA	69.6±NA	-	-	66.7±NA	67.7±NA	66.3±NA	-	-
	EDL	66.9±0.2	71.5±0.2	72.8±0.5	72.2±0.6	77.4±0.1	62.8±0.4	68.9±0.6	71.4±0.7	71.0±0.7	74.8±0.4
	OE	89.5±1.0	91.2±0.9	-	-	-	91.1±0.9	92.6±0.8	-	-	-
	DPN$_{rev}$	81.2±0.2	82.4±0.1	83.8±0.1	83.8±0.1	83.5±0.1	84.7±0.0	86.1±0.0	87.6±0.0	87.6±0.0	87.1±0.0
	DPN$^+$	85.9±0.3	88.1±0.2	92.2±0.1	92.2±0.1	92.3±0.1	88.0±0.2	90.0±0.2	92.7±0.1	92.7±0.1	92.8±0.1
	DPN$^-$	89.2±0.1	90.7±0.1	**94.5±0.1**	**94.5±0.1**	38.1±0.5	91.4±0.1	92.7±0.1	**95.1±0.1**	**95.1±0.1**	55.7±0.4
STL-10	Baseline	69.6±0.0	71.9±0.0	-	-	-	61.9±0.1	65.4±0.1	-	-	-
	MCDP	70.7±0.1	71.2±0.1	71.6±0.2	-	-	62.8±0.1	63.9±0.2	61.4±0.1	-	-
	DE	69.6±NA	70.1±NA	70.2±NA	-	-	62.0±NA	63.0±NA	60.9±NA	-	-
	EDL	68.1±0.2	72.0±0.2	68.0±0.6	67.3±0.7	73.8±0.4	58.5±0.6	64.1±0.4	61.6±1.0	61.1±1.1	66.1±0.7
	OE	91.2±0.7	92.7±0.6	-	-	-	92.1±0.6	93.4±0.5	-	-	-
	DPN$_{rev}$	87.2±0.1	88.1±0.1	89.3±0.1	89.3±0.1	89.0±0.1	88.5±0.0	89.6±0.1	91.0±0.1	91.1±0.1	90.5±0.1
	DPN$^+$	89.1±0.2	90.8±0.2	95.0±0.0	95.0±0.0	94.8±0.0	90.0±0.2	91.7±0.2	94.7±0.0	94.7±0.0	94.6±0.1
	DPN$^-$	92.8±0.1	93.9±0.1	**96.8±0.1**	**96.8±0.1**	25.4±0.4	93.7±0.1	94.7±0.1	**96.7±0.1**	**96.7±0.1**	42.8±0.3
LSUN	Baseline	72.5±0.0	75.0±0.0	-	-	-	69.0±0.1	72.7±0.1	-	-	-
	MCDP	74.5±0.1	75.1±0.1	75.9±0.2	-	-	70.8±0.3	71.9±0.2	70.4±0.2	-	-
	DE	72.6±NA	73.0±NA	73.4±NA	-	-	69.1±NA	70.0±NA	68.6±NA	-	-
	EDL	67.6±0.6	72.3±0.6	72.8±0.6	72.3±0.6	76.7±0.5	62.3±1.0	69.3±1.1	72.9±0.8	72.5±0.9	76.0±0.5
	OE	92.2±0.9	93.7±0.7	-	-	-	93.7±0.7	94.9±0.7	-	-	-
	DPN$_{rev}$	86.7±0.0	87.9±0.0	89.3±0.1	89.3±0.1	88.9±0.1	89.2±0.0	90.5±0.0	92.0±0.0	92.0±0.0	91.5±0.0
	DPN$^+$	90.3±0.3	92.1±0.3	95.0±0.1	95.0±0.1	95.0±0.1	92.0±0.2	93.6±0.2	95.5±0.1	95.5±0.1	95.6±0.1
	DPN$^-$	92.8±0.1	94.0±0.1	**96.5±0.1**	**96.5±0.1**	31.5±0.4	94.3±0.1	95.3±0.1	**97.0±0.1**	96.9±0.1	52.6±0.3
Places365	Baseline	70.7±0.0	73.2±0.0	-	-	-	88.0±0.0	89.6±0.0	-	-	-
	MCDP	72.1±0.1	72.6±0.1	73.4±0.2	-	-	88.6±0.0	89.0±0.0	88.4±0.1	-	-
	DE	70.8±NA	71.2±NA	72.3±NA	-	-	88.1±NA	88.4±NA	88.6±NA	-	-
	EDL	66.8±0.5	71.0±0.6	70.2±0.3	69.7±0.3	74.3±0.4	85.5±0.5	88.4±0.4	89.2±0.2	89.0±0.3	90.6±0.1
	OE	89.3±1.0	90.9±0.9	-	-	-	97.0±0.3	97.5±0.3	-	-	-
	DPN$_{rev}$	83.0±0.1	84.2±0.1	85.8±0.0	85.8±0.1	85.4±0.0	95.1±0.0	95.6±0.0	96.1±0.0	96.1±0.0	95.9±0.0
	DPN$^+$	87.1±0.2	89.3±0.2	92.9±0.1	92.9±0.1	93.0±0.1	96.3±0.0	96.9±0.1	97.9±0.0	97.9±0.0	97.9±0.0
	DPN$^-$	89.9±0.2	91.3±0.1	**94.5±0.1**	**94.5±0.1**	37.9±0.5	97.2±0.0	97.6±0.0	**98.4±0.0**	**98.4±0.0**	79.8±0.2
Textures	Baseline	62.8±0.2	64.7±0.2	-	-	-	43.8±0.2	45.9±0.2	-	-	-
	MCDP	64.3±0.3	64.6±0.2	67.2±0.2	-	-	44.9±0.3	45.3±0.2	49.6±0.0	-	-
	DE	62.9±NA	63.0±NA	69.0±NA	-	-	43.9±NA	44.1±NA	56.1±NA	-	-
	EDL	66.9±0.7	72.1±0.8	73.7±0.5	73.1±0.4	78.1±1.4	47.5±2.2	55.5±2.4	60.1±1.0	59.5±0.9	64.8±1.4
	OE	79.7±1.0	81.2±1.0	-	-	-	71.7±1.5	73.2±1.6	-	-	-
	DPN$_{rev}$	73.7±0.5	75.3±0.4	76.9±0.4	76.9±0.4	76.8±0.4	67.3±0.4	70.5±0.3	73.2±0.3	73.1±0.3	72.9±0.3
	DPN$^+$	78.8±0.1	81.3±0.1	83.6±0.1	83.6±0.1	84.8±0.1	68.5±0.2	71.4±0.1	71.8±0.1	71.8±0.1	73.3±0.1
	DPN$^-$	77.5±0.0	79.5±0.0	**85.2±0.1**	**85.2±0.1**	58.4±0.2	71.9±0.1	74.4±0.1	**78.9±0.2**	**78.9±0.2**	52.1±0.2

Table 13: Results of OOD image detection for TIM. We report (mean $\pm$ standard deviation) of three different models. Description of these OOD datasets are provided in Appendix B.2.1.

	Methods	AUROC					AUPR				
		Max.P	Ent.	MI	α_0	D-Ent	Max.P	Ent.	MI	α_0	D-Ent
CIFAR-10	Baseline	76.9$_{\pm 0.2}$	79.2$_{\pm 0.2}$	-	-	-	72.6$_{\pm 0.2}$	75.1$_{\pm 0.1}$	-	-	-
	MCDP	77.4$_{\pm 0.1}$	79.8$_{\pm 0.1}$	77.5$_{\pm 0.2}$	-	-	73.0$_{\pm 0.2}$	75.4$_{\pm 0.1}$	71.4$_{\pm 0.4}$	-	-
	DE	76.9$_{\pm \text{NA}}$	79.3$_{\pm \text{NA}}$	77.7$_{\pm \text{NA}}$	-	-	72.6$_{\pm \text{NA}}$	75.2$_{\pm \text{NA}}$	72.4$_{\pm \text{NA}}$	-	-
	EDL	74.4$_{\pm 0.7}$	76.1$_{\pm 0.7}$	72.0$_{\pm 1.3}$	71.8$_{\pm 1.3}$	73.3$_{\pm 1.1}$	69.4$_{\pm 0.3}$	70.9$_{\pm 0.8}$	67.8$_{\pm 1.3}$	67.7$_{\pm 1.4}$	68.7$_{\pm 1.3}$
	OE	91.3$_{\pm 0.4}$	92.6$_{\pm 0.3}$	-	-	-	92.9$_{\pm 0.4}$	94.0$_{\pm 0.3}$	-	-	-
	DPN$_{rev}$	85.4$_{\pm 0.7}$	86.0$_{\pm 0.8}$	82.8$_{\pm 1.4}$	81.9$_{\pm 1.6}$	85.6$_{\pm 0.9}$	88.1$_{\pm 0.7}$	88.7$_{\pm 0.7}$	86.5$_{\pm 1.3}$	86.0$_{\pm 1.4}$	88.7$_{\pm 0.8}$
	DPN$^+$	99.2$_{\pm 0.0}$	99.4$_{\pm 0.0}$	99.7$_{\pm 0.0}$	99.7$_{\pm 0.0}$	99.6$_{\pm 0.0}$	99.4$_{\pm 0.0}$	99.5$_{\pm 0.0}$	99.7$_{\pm 0.0}$	99.7$_{\pm 0.0}$	99.6$_{\pm 0.0}$
	DPN$^-$	99.7$_{\pm 0.0}$	99.8$_{\pm 0.0}$	**99.9$_{\pm 0.0}$**	**99.9$_{\pm 0.0}$**	3.5$_{\pm 0.1}$	99.8$_{\pm 0.0}$	99.8$_{\pm 0.0}$	**99.9$_{\pm 0.0}$**	**99.9$_{\pm 0.0}$**	33.6$_{\pm 0.1}$
CIFAR-100	Baseline	73.6$_{\pm 0.2}$	75.7$_{\pm 0.2}$	-	-	-	69.7$_{\pm 0.2}$	72.2$_{\pm 0.2}$	-	-	-
	MCDP	74.0$_{\pm 0.2}$	76.1$_{\pm 0.2}$	73.6$_{\pm 0.2}$	-	-	70.0$_{\pm 0.1}$	72.3$_{\pm 0.1}$	67.6$_{\pm 0.3}$	-	-
	DE	73.7$_{\pm \text{NA}}$	75.8$_{\pm \text{NA}}$	75.3$_{\pm \text{NA}}$	-	-	69.7$_{\pm \text{NA}}$	72.3$_{\pm \text{NA}}$	70.6$_{\pm \text{NA}}$	-	-
	EDL	71.8$_{\pm 0.6}$	73.7$_{\pm 0.7}$	70.1$_{\pm 0.9}$	69.9$_{\pm 0.9}$	71.4$_{\pm 0.8}$	67.2$_{\pm 0.4}$	68.6$_{\pm 0.9}$	65.9$_{\pm 1.0}$	65.8$_{\pm 1.0}$	66.7$_{\pm 1.0}$
	OE	89.5$_{\pm 0.5}$	90.9$_{\pm 0.4}$	-	-	-	91.5$_{\pm 0.5}$	92.7$_{\pm 0.4}$	-	-	-
	DPN$_{rev}$	84.2$_{\pm 0.8}$	85.0$_{\pm 0.8}$	82.5$_{\pm 1.4}$	81.7$_{\pm 1.6}$	85.0$_{\pm 0.9}$	87.0$_{\pm 0.7}$	87.9$_{\pm 0.7}$	86.3$_{\pm 1.3}$	85.7$_{\pm 1.4}$	88.2$_{\pm 0.8}$
	DPN$^+$	98.8$_{\pm 0.0}$	99.0$_{\pm 0.0}$	99.5$_{\pm 0.0}$	99.5$_{\pm 0.0}$	99.4$_{\pm 0.0}$	99.1$_{\pm 0.0}$	99.3$_{\pm 0.0}$	99.5$_{\pm 0.0}$	99.5$_{\pm 0.0}$	99.5$_{\pm 0.0}$
	DPN$^-$	98.7$_{\pm 0.1}$	99.0$_{\pm 0.0}$	**99.6$_{\pm 0.0}$**	**99.6$_{\pm 0.0}$**	7.5$_{\pm 0.2}$	99.1$_{\pm 0.0}$	99.3$_{\pm 0.0}$	**99.7$_{\pm 0.0}$**	**99.7$_{\pm 0.0}$**	36.7$_{\pm 0.2}$
STL-10	Baseline	67.7$_{\pm 0.1}$	67.7$_{\pm 0.1}$	-	-	-	56.5$_{\pm 0.1}$	56.8$_{\pm 0.2}$	-	-	-
	MCDP	68.1$_{\pm 0.1}$	68.1$_{\pm 0.1}$	69.8$_{\pm 0.1}$	-	-	56.8$_{\pm 0.2}$	57.3$_{\pm 0.2}$	60.1$_{\pm 0.4}$	-	-
	DE	67.7$_{\pm \text{NA}}$	67.8$_{\pm \text{NA}}$	68.1$_{\pm \text{NA}}$	-	-	56.5$_{\pm \text{NA}}$	56.8$_{\pm \text{NA}}$	58.1$_{\pm \text{NA}}$	-	-
	EDL	66.5$_{\pm 0.3}$	65.0$_{\pm 0.6}$	60.4$_{\pm 1.0}$	60.3$_{\pm 1.0}$	61.3$_{\pm 0.9}$	55.7$_{\pm 0.1}$	55.6$_{\pm 0.4}$	53.7$_{\pm 0.7}$	53.6$_{\pm 0.8}$	54.1$_{\pm 0.7}$
	OE	**100.0$_{\pm 0.0}$**	**100.0$_{\pm 0.0}$**	-	-	-	**100.0$_{\pm 0.0}$**	**100.0$_{\pm 0.0}$**	-	-	-
	DPN$_{rev}$	99.4$_{\pm 0.0}$	99.5$_{\pm 0.0}$	99.6$_{\pm 0.0}$	99.6$_{\pm 0.0}$	99.5$_{\pm 0.0}$	99.5$_{\pm 0.0}$	99.6$_{\pm 0.0}$	99.7$_{\pm 0.0}$	99.7$_{\pm 0.0}$	99.6$_{\pm 0.0}$
	DPN$^+$	99.9$_{\pm 0.0}$	**100.0$_{\pm 0.0}$**	**100.0$_{\pm 0.0}$**	**100.0$_{\pm 0.0}$**	**100.0$_{\pm 0.0}$**	99.9$_{\pm 0.0}$	**100.0$_{\pm 0.0}$**	99.9$_{\pm 0.1}$	99.9$_{\pm 0.1}$	99.9$_{\pm 0.0}$
	DPN$^-$	**100.0$_{\pm 0.0}$**	**100.0$_{\pm 0.0}$**	**100.0$_{\pm 0.0}$**	**100.0$_{\pm 0.0}$**	0.2$_{\pm 0.0}$	**100.0$_{\pm 0.0}$**	**100.0$_{\pm 0.0}$**	**100.0$_{\pm 0.0}$**	**100.0$_{\pm 0.0}$**	26.6$_{\pm 0.0}$
LSUN	Baseline	69.0$_{\pm 0.0}$	70.2$_{\pm 0.1}$	-	-	-	63.5$_{\pm 0.2}$	64.0$_{\pm 0.3}$	-	-	-
	MCDP	69.3$_{\pm 0.1}$	70.4$_{\pm 0.1}$	69.5$_{\pm 0.2}$	-	-	63.6$_{\pm 0.2}$	64.0$_{\pm 0.3}$	63.2$_{\pm 0.4}$	-	-
	DE	69.1$_{\pm \text{NA}}$	70.2$_{\pm \text{NA}}$	70.3$_{\pm \text{NA}}$	-	-	63.5$_{\pm \text{NA}}$	64.0$_{\pm \text{NA}}$	65.2$_{\pm \text{NA}}$	-	-
	EDL	65.5$_{\pm 0.2}$	67.1$_{\pm 0.6}$	64.7$_{\pm 1.3}$	64.7$_{\pm 1.3}$	65.6$_{\pm 1.2}$	60.4$_{\pm 0.3}$	61.8$_{\pm 0.8}$	60.6$_{\pm 1.5}$	60.5$_{\pm 1.6}$	61.0$_{\pm 1.4}$
	OE	**100.0$_{\pm 0.0}$**	**100.0$_{\pm 0.0}$**	-	-	-	**100.0$_{\pm 0.0}$**	**100.0$_{\pm 0.0}$**	-	-	-
	DPN$_{rev}$	99.6$_{\pm 0.0}$	99.7$_{\pm 0.0}$	99.8$_{\pm 0.0}$	99.8$_{\pm 0.0}$	99.8$_{\pm 0.0}$	99.7$_{\pm 0.0}$	99.8$_{\pm 0.0}$	99.8$_{\pm 0.0}$	99.8$_{\pm 0.0}$	99.8$_{\pm 0.0}$
	DPN$^+$	**100.0$_{\pm 0.0}$**	**100.0$_{\pm 0.0}$**	**100.0$_{\pm 0.0}$**	**100.0$_{\pm 0.0}$**	**100.0$_{\pm 0.0}$**	**100.0$_{\pm 0.0}$**	**100.0$_{\pm 0.0}$**	99.9$_{\pm 0.0}$	99.9$_{\pm 0.0}$	99.9$_{\pm 0.0}$
	DPN$^-$	**100.0$_{\pm 0.0}$**	**100.0$_{\pm 0.0}$**	**100.0$_{\pm 0.0}$**	**100.0$_{\pm 0.0}$**	0.2$_{\pm 0.0}$	**100.0$_{\pm 0.0}$**	**100.0$_{\pm 0.0}$**	**100.0$_{\pm 0.0}$**	**100.0$_{\pm 0.0}$**	30.8$_{\pm 0.0}$
Places365	Baseline	71.1$_{\pm 0.1}$	72.7$_{\pm 0.1}$	-	-	-	87.5$_{\pm 0.0}$	88.2$_{\pm 0.0}$	-	-	-
	MCDP	71.4$_{\pm 0.1}$	73.0$_{\pm 0.0}$	70.8$_{\pm 0.1}$	-	-	87.7$_{\pm 0.0}$	88.2$_{\pm 0.0}$	86.8$_{\pm 0.2}$	-	-
	DE	71.2$_{\pm \text{NA}}$	72.7$_{\pm \text{NA}}$	71.6$_{\pm \text{NA}}$	-	-	87.6$_{\pm \text{NA}}$	88.2$_{\pm \text{NA}}$	87.6$_{\pm \text{NA}}$	-	-
	EDL	69.0$_{\pm 0.1}$	70.5$_{\pm 0.2}$	67.1$_{\pm 0.4}$	66.9$_{\pm 0.4}$	68.2$_{\pm 0.4}$	86.5$_{\pm 0.2}$	87.1$_{\pm 0.2}$	86.0$_{\pm 0.3}$	86.0$_{\pm 0.3}$	86.4$_{\pm 0.3}$
	OE	99.9$_{\pm 0.0}$	99.9$_{\pm 0.0}$	-	-	-	**100.0$_{\pm 0.0}$**	**100.0$_{\pm 0.0}$**	-	-	-
	DPN$_{rev}$	98.9$_{\pm 0.1}$	99.1$_{\pm 0.0}$	99.2$_{\pm 0.1}$	99.2$_{\pm 0.1}$	99.2$_{\pm 0.0}$	99.7$_{\pm 0.0}$	99.8$_{\pm 0.0}$	99.8$_{\pm 0.0}$	99.8$_{\pm 0.0}$	99.8$_{\pm 0.0}$
	DPN$^+$	99.9$_{\pm 0.0}$	99.9$_{\pm 0.0}$	99.9$_{\pm 0.0}$	99.9$_{\pm 0.0}$	99.9$_{\pm 0.0}$	**100.0$_{\pm 0.0}$**	**100.0$_{\pm 0.0}$**	**100.0$_{\pm 0.0}$**	**100.0$_{\pm 0.0}$**	**100.0$_{\pm 0.0}$**
	DPN$^-$	99.9$_{\pm 0.0}$	99.9$_{\pm 0.0}$	**100.0$_{\pm 0.0}$**	**100.0$_{\pm 0.0}$**	0.8$_{\pm 0.0}$	**100.0$_{\pm 0.0}$**	**100.0$_{\pm 0.0}$**	**100.0$_{\pm 0.0}$**	**100.0$_{\pm 0.0}$**	58.5$_{\pm 0.0}$
Textures	Baseline	70.9$_{\pm 0.2}$	73.3$_{\pm 0.2}$	-	-	-	53.8$_{\pm 0.4}$	57.5$_{\pm 0.5}$	-	-	-
	MCDP	70.3$_{\pm 0.2}$	72.6$_{\pm 0.3}$	63.6$_{\pm 0.2}$	-	-	53.1$_{\pm 0.3}$	56.5$_{\pm 0.5}$	41.6$_{\pm 0.2}$	-	-
	DE	71.1$_{\pm \text{NA}}$	73.4$_{\pm \text{NA}}$	76.2$_{\pm \text{NA}}$	-	-	53.9$_{\pm \text{NA}}$	57.7$_{\pm \text{NA}}$	62.2$_{\pm \text{NA}}$	-	-
	EDL	71.0$_{\pm 0.1}$	74.0$_{\pm 0.4}$	70.4$_{\pm 0.9}$	70.2$_{\pm 0.9}$	71.9$_{\pm 0.9}$	52.8$_{\pm 0.5}$	55.8$_{\pm 0.4}$	54.3$_{\pm 0.9}$	54.2$_{\pm 0.9}$	54.9$_{\pm 0.8}$
	OE	95.8$_{\pm 0.3}$	96.4$_{\pm 0.3}$	-	-	-	95.5$_{\pm 0.3}$	96.1$_{\pm 0.3}$	-	-	-
	DPN$_{rev}$	90.9$_{\pm 0.3}$	91.9$_{\pm 0.3}$	91.2$_{\pm 0.6}$	90.6$_{\pm 0.6}$	92.6$_{\pm 0.3}$	89.9$_{\pm 0.3}$	91.3$_{\pm 0.3}$	90.7$_{\pm 0.6}$	90.2$_{\pm 0.6}$	92.3$_{\pm 0.3}$
	DPN$^+$	96.5$_{\pm 0.1}$	97.1$_{\pm 0.0}$	98.4$_{\pm 0.0}$	98.4$_{\pm 0.0}$	98.2$_{\pm 0.0}$	96.2$_{\pm 0.0}$	96.9$_{\pm 0.0}$	97.9$_{\pm 0.0}$	97.9$_{\pm 0.0}$	97.8$_{\pm 0.0}$
	DPN$^-$	95.8$_{\pm 0.1}$	96.8$_{\pm 0.1}$	**98.7$_{\pm 0.1}$**	**98.7$_{\pm 0.1}$**	19.3$_{\pm 0.4}$	95.4$_{\pm 0.0}$	96.5$_{\pm 0.0}$	**98.5$_{\pm 0.1}$**	**98.5$_{\pm 0.1}$**	35.7$_{\pm 0.3}$