**Abstract**

Ensembles of models often yield improvements in system performance. These ensemble approaches have also been empirically shown to yield robust measures of uncertainty, and are capable of distinguishing between different forms of uncertainty. However, ensembles come at a computational and memory cost which may be prohibitive for many applications. There has been significant work done on the distillation of an ensemble into a single model. Such approaches decrease computational cost and allow a single model to achieve an accuracy comparable to that of an ensemble. However, information about the diversity of the ensemble, which can yield estimates of different forms of uncertainty, is lost. This work considers the novel task of *Ensemble Distribution Distillation* (EnD$^2$) — distilling the distribution of the predictions from an ensemble, rather than just the average prediction, into a single model. EnD$^2$ enables a single model to retain both the improved classification performance of ensemble distillation as well as information about the diversity of the ensemble, which is useful for uncertainty estimation. A solution for EnD$^2$ based on Prior Networks, a class of models which allow a single neural network to explicitly model a distribution over output distributions, is proposed in this work. The properties of EnD$^2$ are investigated on both an artificial dataset, and on the CIFAR-10, CIFAR-100 and TinyImageNet datasets, where it is shown that EnD$^2$ can approach the classification performance of an ensemble, and outperforms both standard DNNs and Ensemble Distillation on the tasks of misclassification and out-of-distribution input detection.

**1 Introduction**

Neural Networks (NNs) have emerged as the state-of-the-art approach to a variety of machine learning tasks ([LeCun et al., 2015](#)) in domains such as computer vision ([Girshick, 2015](#)), [Simonyan & Zisserman, 2015](#), [Villegas et al., 2017](#), natural language processing ([Mikolov et al., 2013b; 2010](#)), speech recognition ([Hinton et al., 2012](#)), [Hannun et al., 2014](#) and bio-informatics ([Caruana et al., 2015](#), [Alipanahi et al., 2015](#)). Despite impressive supervised learning performance, NNs tend to make over-confident predictions ([Lakshminarayanan et al., 2017](#)) and, until recently, have been unable to provide measures of uncertainty in their predictions. As NNs are increasingly being applied to safety-critical tasks such as medical diagnosis ([De Fauw et al., 2018](#)), biometric identification ([Schroff et al., 2015](#)) and self driving cars, estimating uncertainty in model’s predictions is crucial, as it enables the safety of an AI system ([Amodei et al., 2016](#)) to be improved by acting on the predictions in an informed manner.

Ensembles of NNs are known to yield increased accuracy over a single model ([Murphy, 2012](#)), allow useful measures of uncertainty to be derived ([Lakshminarayanan et al., 2017](#)), and also provide defense against adversarial attacks ([Smith & Gal, 2018](#)). There is both a range of Bayesian Monte-Carlo approaches ([Gal & Ghahramani, 2016](#), [Welling & Teh, 2011](#), [Garipov et al., 2018](#), [Maddox et al., 2019](#)), as well as non-Bayesian approaches, such as random-initialization ([Lakshminarayanan et al., 2017](#)) and bagging ([Murphy, 2012](#), [Osband et al., 2016](#)), to generating ensembles. Crucially, ensemble approaches allow total uncertainty in predictions to be decomposed into knowledge uncertainty and data uncertainty. Data uncertainty is the irreducible uncertainty in predictions which arises due to the complexity, multi-modality and noise in the data. Knowledge uncertainty, also known as epistemic uncertainty ([Gal, 2016](#) or distributional uncertainty ([Malinin & Gales, 2018](#)), is uncertainty due to a lack of understanding or knowledge on the part of the model regarding the current input for which the model is making a prediction. This form of uncertainty arises when the test input $x^*$ comes either...
from a different distribution than the one that generated the training data or from an in-domain region which is sparsely covered by the training data. Mismatch between the test and training distributions is also known as a dataset shift (Quinonero-Candela, 2009), and is a situation which often arises for real world problems. Distinguishing between sources of uncertainty is important, as in certain machine learning applications it may be necessary to know not only whether the model is uncertain, but also why. For instance, in active learning, additional training data should be collected from regions with high knowledge uncertainty, but not data uncertainty.

A fundamental limitation of ensembles is that the computational cost of training and, more importantly, inference can be many times greater than that of a single model. One solution to speed up inference is to distil an ensemble of models into a single network to yield the mean predictions of the ensemble (Hinton et al., 2015; Korattikara Balan et al., 2015). However, this collapses an ensemble of conditional distributions over classes into a single point-estimate conditional distribution over classes. As a result, information about the diversity of the ensemble is lost. This prevents measures of knowledge uncertainty, such as mutual information (Malinin & Gales, 2018; Depeweg et al., 2017a), from being estimated.

In this work, we investigate the explicit modelling of the distribution over the ensemble predictions, rather than just the mean, with a single model. This problem — referred to as Ensemble Distribution Distillation (EnD²) — yields a method that preserves both the distributional information and improved classification performance of an ensemble within a single neural network model. It is important to highlight that Ensemble Distribution Distillation is a novel task which, to our knowledge, has not been previously investigated. Here, the goal is to extract as much information as possible from an ensemble of models and retain it within a single, possibly simpler, model. As an initial solution to this problem, this paper makes use of a recently introduced class of models, known as Prior Networks (Malinin & Gales, 2018; 2019), which explicitly model a conditional distribution over categorical distributions by parameterizing a Dirichlet distribution. Within the context of EnD² this effectively allows a single model to emulate the complete ensemble.

The contributions of this work are as follows. Firstly, we define the task of Ensemble Distribution Distillation (EnD²) as a new challenge for machine learning research. Secondly, we propose and evaluate a solution to this problem using Prior Networks. EnD² is initially investigated on artificial data, which allows the behaviour of the models to be visualized. It is shown that distribution-distilled models are able to distinguish between data uncertainty and knowledge uncertainty. Finally, EnD² is evaluated on CIFAR-10, CIFAR-100 and TinyImageNet datasets, where it is shown that EnD² yields models which approach the classification performance of the original ensemble and outperform standard DNNs and regular Ensemble Distillation (EnD) models on the tasks of identifying misclassifications and out-of-distribution (OOD) samples.

2 Ensembles

In this work, a Bayesian viewpoint on ensembles is adopted, as it provides a particularly elegant probabilistic framework, which allows knowledge uncertainty to be linked to Bayesian model uncertainty. However, it is also possible to construct ensembles using a range of non-Bayesian approaches. For example, it is possible to explicitly construct an ensemble of M models by training on the same data with different random seeds (Lakshminarayanan et al., 2017) and/or different model architectures. Alternatively, it is possible to generate ensembles via Bootstrap methods (Murphy, 2012; Osband et al., 2016) in which each model is trained on a re-sampled version of the training data.

The essence of Bayesian methods is to treat the model parameters \( \theta \) as random variables and place a prior distribution \( p(\theta) \) over them to compute a posterior distribution \( p(\theta | D) \) via Bayes’ rule:

\[
p(\theta | D) = \frac{p(D | \theta) p(\theta)}{p(D)} \propto p(D | \theta) p(\theta) \tag{1}
\]

Here, model uncertainty is captured in the posterior distribution \( p(\theta | D) \). Consider an ensemble of models \( \{ P(y | x^*, \theta^{(m)}) \}_{m=1}^{M} \) sampled from the posterior:

\[
\{ P(y | x^*, \theta^{(m)}) \}_{m=1}^{M} \rightarrow \{ P(y | \pi^{(m)}) \}_{m=1}^{M}, \quad \pi^{(m)} = f(x^*; \theta^{(m)}), \quad \theta^{(m)} \sim p(\theta | D) \tag{2}
\]

where \( \pi \) are the parameters of a categorical distribution \( [ P(y = \omega_1), \cdots, P(y = \omega_K) ]^T \) and \( x^* \) is a test input. The expected predictive distribution, or predictive posterior, for a test input \( x^* \) is obtained
by taking the expectation with respect to the model posterior:

$$P(y|x^*, D) = \mathbb{E}_p(\theta|D)\left[P(y|x^*, \theta)\right]$$  \hspace{1cm} (3)

Each of the models $P(y|x^*, \theta^{(m)})$ yields a different estimate of data uncertainty. Uncertainty in predictions due to model uncertainty is expressed as the level of spread, or ‘disagreement’, of an ensemble sampled from the posterior. The aim is to craft a posterior $p(\theta|D)$, via an appropriate choice of prior $p(\theta)$, which yields an ensemble that exhibits the set of behaviours described in figure[1]. Specifically, for an in-domain test input $x^*$, the ensemble should produce a consistent set of predictions with little spread, as described in figure[1a] and figure[1b]. In other words, the models should agree in their estimates of data uncertainty. On the other hand, for inputs which are different from the training data, the models in the ensemble should ‘disagree’ and produce a diverse set of predictions, as shown in figure[1c]. Ideally, the models should yield increasingly diverse predictions as input $x^*$ moves further away from the training data. If an input is completely unlike the training data, then the level of disagreement should be significant. Hence, the measures of model uncertainty will capture knowledge uncertainty given an appropriate choice of prior.

Given an ensemble $\{P(y|x^*, \theta^{(m)})\}_{m=1}^{M}$ which exhibits the desired set of behaviours, the entropy of the expected distribution $P(y|x^*, D)$ can be used as a measure of total uncertainty in the prediction. Uncertainty in predictions due to knowledge uncertainty can be assessed via measures of the spread, or ‘disagreement’, of the ensemble such as Mutual Information:

$$\mathcal{MI}_{y, \theta|x^*, D} = \mathcal{H}[\mathbb{E}_p(\theta|D)P(y|x^*, \theta)] - \mathbb{E}_p(\theta|D)\left[\mathcal{H}[P(y|x^*, \theta)]\right]$$  \hspace{1cm} (4)

This formulation of mutual information allows the total uncertainty to be decomposed into knowledge uncertainty and expected data uncertainty (Depeweg et al., 2017a,b). The entropy of the predictive posterior, or total uncertainty, will be high whenever the model is uncertain - both in regions of severe class overlap and out-of-domain. However, the difference of the entropy of the predictive posterior and the expected entropy of the individual models will be non-zero only if the models disagree. For example, in regions of class overlap, each member of the ensemble will yield a high entropy distribution (figure[1b]) - the entropy of the predictive posterior and the expected entropy will be similar and mutual information will be low. In this situation total uncertainty is dominated by data uncertainty. On the other hand, for out-of-domain inputs the ensemble yields diverse distributions over classes such that the predictive posterior is near uniform (figure[1c]), while the expected entropy of each model may be much lower. In this region of input space the models’ understanding of data is low and, therefore, knowledge uncertainty is high.

3 Ensemble Distribution Distillation

Previous work (Hinton et al., 2015; Korattikara Balan et al., 2015; Wong & Gales, 2017; Wang et al., 2018; Papamakarios, 2015; Bucilua et al., 2006) has investigated distilling a single large network into a smaller one and an ensemble of networks into a single neural network. In general, distillation is done by minimizing the KL-divergence between the model and the expected predictive distribution of an ensemble:

$$\mathcal{L}(\phi, D_{\text{ens}}) = \mathbb{E}_p(x)\left[\text{KL}[\mathbb{E}_p(\theta|D)P(y|x; \theta) || P(y|x; \phi)]\right]$$  \hspace{1cm} (5)
This approach essentially aims to train a single model that captures the mean of an ensemble, allowing the model to achieve a higher classification performance at a far lower computational cost. The use of such models for uncertainty estimation was investigated in (Li & Hoiem, 2019, Englesson & Azizpour, 2019). However, the limitation of this approach with regards to uncertainty estimation is that the information about the diversity of the ensemble is lost. As a result, it is no longer possible to decompose total uncertainty into knowledge uncertainty and data uncertainty via mutual information as in equation 4. In this work we propose the task of Ensemble Distribution Distillation, where the goal is to capture not only the mean of the ensemble, but also its diversity. In this section, we outline an initial solution to this task.

An ensemble can be viewed as a set of samples from an implicit distribution of output distributions:

\[
\{ p(y|x^*, \theta^{(m)}) \}_{m=1}^{M} \to \{ p(y|\pi^{(m)}) \}_{m=1}^{M}, \quad \pi^{(m)} \sim p(\pi|x^*, D) \]  

(6)

Recently, a new class of models was proposed, called Prior Networks (Malinin & Gales, 2018, 2019), which explicitly parameterize a conditional distribution over output distributions \( p(\pi|x^*; \tilde{\phi}) \) using a single neural network parameterized by a point estimate of the model parameters \( \tilde{\phi} \). Thus, a Prior Network is able to effectively emulate an ensemble, and therefore yield the same measures of uncertainty. A Prior Network \( p(\pi|x^*; \tilde{\phi}) \) models a distribution over categorical output distributions by parameterizing the Dirichlet distribution.

\[
p(\pi|x; \tilde{\phi}) = \text{Dir}(\pi|\hat{\alpha}), \quad \hat{\alpha} = f(x; \tilde{\phi}), \quad \hat{\alpha}_c > 0, \quad \hat{\alpha}_0 = \sum_{c=1}^{K} \hat{\alpha}_c \]  

(7)

The distribution is parameterized by its concentration parameters \( \alpha \), which can be obtained by placing an exponential function at the output of a Prior Network: \( \hat{\alpha}_c = e^{\tilde{\alpha}_c} \), where \( \tilde{\alpha} \) are the logits predicted by the model. While a Prior Network could, in general, parameterize arbitrary distributions over categorical distributions, the Dirichlet is chosen due to its tractable analytic properties, which allow closed form expressions for all measures of uncertainty to be obtained. However, it is important to note that the Dirichlet distribution may be too limited to fully capture the behaviour of an ensemble and other distributions may need to be considered.

In this work we consider how an ensemble, which is a set of samples from an implicit distribution over distributions, can be distribution distilled into an explicit distribution over distributions modelled using a single Prior Network model, ie: \( \{ p(y|x; \theta^{(m)}) \}_{m=1}^{M} \to p(\pi|x; \tilde{\phi}) \).

This is accomplished in several steps. Firstly, a transfer dataset \( D_{\text{ens}} = \{ x^{(i)}, \pi^{(1:1:M)} \}_{i=1}^{N} \sim p(x, \pi) \) is composed of the inputs \( x \) from the original training set \( D = \{ x^{(i)}, y^{(i)} \}_{i=1}^{N} \) and the categorical distributions \{ \pi^{(1:1:M)} \}_{i=1}^{N} \) derived from the ensemble for each input. Secondly, given this transfer set, the model \( p(\pi|x; \tilde{\phi}) \) is trained by minimizing the negative log-likelihood of each categorical distribution \( \pi^{(im)} \):

\[
\mathcal{L}(\phi, D_{\text{ens}}) = -\mathbb{E}_{p(x)} \left[ \mathbb{E}_{p(\pi|x)} [\ln p(\pi|x; \phi)] \right] \\
= -\frac{1}{N} \sum_{i=1}^{N} \left[ \ln \Gamma(\hat{\alpha}_0) - \sum_{c=1}^{K} \ln \Gamma(\hat{\alpha}_c) + \frac{1}{M} \sum_{m=1}^{K} \sum_{c=1}^{K} (\hat{\alpha}_c - 1) \ln \pi_c^{(im)} \right] 
\]  

(8)

Thus, Ensemble Distribution Distillation with Prior Networks is a straightforward application of maximum-likelihood estimation. Given a distribution-distilled Prior Network, the predictive distribution is given by the expected categorical distribution \( \bar{\pi} \) under the Dirichlet prior:

\[
P(y = \omega_c | x^*, \tilde{\phi}) = \mathbb{E}_{p(\pi|x^*, \tilde{\phi})} [p(y = \omega_c | \pi)] = \bar{\pi}_c = \frac{\hat{\alpha}_c}{\sum_{k=1}^{K} \hat{\alpha}_k} = \frac{e^{\tilde{\alpha}_c}}{\sum_{k=1}^{K} e^{\tilde{\alpha}_k}} 
\]  

(9)

Separable measures of uncertainty can be obtained by considering the mutual information between the prediction \( y \) and the parameters of \( \pi \) of the categorical:

\[
\text{MI}(y, \pi|x^*, \tilde{\phi}) = \mathcal{H}[\mathbb{E}_{p(\pi|x^*, \tilde{\phi})} [p(y|\pi)]] - \mathbb{E}_{p(\pi|x^*, \tilde{\phi})} [\mathcal{H}(p(y|\pi))] 
\]  

(10)

Similar to equation 4, this expression allows total uncertainty, given by the entropy of the expected distribution, to be decomposed into data uncertainty and knowledge uncertainty (Malinin & Gales, 2018). If Ensemble Distribution Distillation is successful, then the measures of uncertainty derivable from a distribution-distilled model should be identical to those derived from the original ensemble.
3.1 Temperature Annealing

Minimization of the negative log-likelihood of the model on the transfer dataset $\mathcal{D}_{\text{ens}} = \{x^{(i)}, \pi^{(i,1:M)}\}_{i=1}^{N}$ is equivalent to minimization of the KL-divergence between the model and the empirical distribution $\hat{p}(x, \pi)$. On training data, this distribution is often ‘sharp’ at one of the corners of the simplex. At the same time, the Dirichlet distribution predicted by the model has its mode near the center of the simplex with little support at the corners at initialization. Thus, the common support between the model and the target empirical distribution is limited. Optimization of the KL-divergence between distributions with limited non-zero common support is particularly difficult. To alleviate this issue, and improve convergence, the proposed solution is to use temperature to ‘heat up’ both distributions and increase common support by moving the modes of both distributions closer together. The empirical distribution is ‘heated up’ by raising the temperature $T$ of the softmax of each model in the ensemble in the same way as in (Hinton et al., 2015). This moves the predictions of the ensemble closer to the center of the simplex and decreases their diversity, making it better modelled by a sharp Dirichlet distribution. The output distribution of the EnD$^2$ model $p(\pi|x; \phi)$ is heated up by raising the temperature of the concentration parameters: $\hat{\alpha}_c = e^{\hat{z}_c}/T$, making support more uniform across the simplex. An annealing schedule is used to re-emphasize the diversity of the empirical distribution and return it to its ‘natural’ state by lowering the temperature down to 1 as training progresses.

4 Experiments on Artificial Data

![Figure 2: 3-spiral dataset with 1000 examples per class](image)

The current section investigates Ensemble Distribution Distillation (EnD$^2$) on an artificial dataset shown in figure 2a. This dataset consists of three spiral arms extending from the center with both increasing noise and distance between the arms. Each arm corresponds to a single class. This dataset is chosen such that it is not linearly separable and requires a powerful model to correctly model the decision boundaries, and also such that there are definite regions of class overlap.

In the following set of experiments, an ensemble of 100 neural networks is constructed by training neural networks from 100 different random initializations. A smaller (sub) ensemble of only 10 neural networks is also considered. The models are trained on 3000 data-points sampled from the spiral dataset, with 1000 examples per class. The classification performance of EnD$^2$ is compared to the performance of individual neural networks, the overall ensemble and regular Ensemble Distillation (EnD). The results are presented in table 1.

<table>
<thead>
<tr>
<th>Num. models</th>
<th>Individual</th>
<th>Ensemble</th>
<th>EnD</th>
<th>EnD$^2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>13.21</td>
<td>12.63</td>
<td>12.57</td>
<td>12.52</td>
</tr>
<tr>
<td>100</td>
<td>12.37</td>
<td>12.39</td>
<td>12.47</td>
<td></td>
</tr>
</tbody>
</table>

The results show that an ensemble of 10 models has a clear performance gain compared to the mean performance of the individual models. An ensemble of 100 models has a smaller performance gain
over an ensemble of only 10 models. Ensemble Distillation (EnD) is able to recover the classification performance of both an ensemble of 10 and 100 models with only very minor degradation in performance. Finally, Ensemble Distribution Distillation is also able to recover most of the performance gain of an ensemble, but with a slightly larger degradation. This is likely due to forcing a single model to learn not only the mean, but also the distribution around it, which likely requires more capacity from the network. The measures of uncertainty derived from an ensemble of 100 models and

![Figure 3: Comparison of measures of uncertainty derived from an Ensemble, EnD^2 and EnD^2+\text{AUX}.](image)

from Ensemble Distribution Distillation are presented in figures 3a-c and figures 3d-f, respectively. The results show that EnD^2 successfully captures data uncertainty and also correctly decomposes total uncertainty into knowledge uncertainty and data uncertainty. However, it fails to appropriately capture knowledge uncertainty further away from the training region, as there are obvious dark holes in figure 3f, where the model yields low knowledge uncertainty far from the region of training data.

In order to overcome these issues, a thick ring of inputs far from the training data was sampled as depicted in figure 2b. The predictions of the ensemble were obtained for these input points and used as additional auxiliary training data \( D_{\text{AUX}} \). Table 2 shows how using the auxiliary training data affects the performance of the Ensemble Distillation and Ensemble Distribution Distillation. There is a minor drop in performance of both distillation approaches. However, the overall level of performance is not compromised and is still higher than the average performance of each individual DNN model.

The behaviour of measures of uncertainty derived from Ensemble Distribution Distillation with auxiliary training data (EnD^2_{\text{AUX}}) is shown in figures 3g-i. These results show that successful Ensemble Distribution Distillation of the out-of-distribution behaviour of an ensemble based purely on observations of the in-domain behaviour is challenging and may require the use of additional
training data. This is compounded by the fact that the diversity of an ensemble on training data that the model has seen is typically smaller than on a heldout test-set.

Table 2: Classification Performance (% Error) on \( D_{\text{test}} \), trained on either \( D_{\text{trn}} \) or \( D_{\text{trn}} + D_{\text{AUX}} \). All datasets are of size 1000. Data for an ensemble of a 100 models.

<table>
<thead>
<tr>
<th>Distillation Data</th>
<th>Individual</th>
<th>Ensemble</th>
<th>EnD</th>
<th>EnD^2</th>
</tr>
</thead>
<tbody>
<tr>
<td>( D_{\text{trn}} )</td>
<td>13.21</td>
<td>12.37</td>
<td>12.39</td>
<td>12.47</td>
</tr>
<tr>
<td>( D_{\text{trn}} + D_{\text{AUX}} )</td>
<td>12.41</td>
<td>12.50</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

5 Experiments on Image Data

Having confirmed the properties of EnD^2 on an artificial dataset, we now investigate Ensemble Distribution Distillation on the CIFAR-10 (C10), CIFAR-100 (C100) and TinyImageNet (TIM) datasets. Similarly to section 4, an ensemble of a 100 models is constructed by training NNs on C10/100/TIM data from different random initializations. The transfer dataset is constructed from C10/100/TIM inputs and ensemble logits to allow for temperature annealing during training, which we found to be essential to getting EnD^2 to train well. In addition, we also consider Ensemble Distillation and Ensemble Distribution Distillation on a transfer set that contains both the original C10/C100/TIM training data and auxiliary (AUX) data taken from the other dataset. It is important to note that the auxiliary data has been treated in the same way as main data during construction of the transfer set and distillation. This offers an advantage over traditional Prior Network training (Malinin & Gales, 2018; 2019), where the knowledge of which examples are in-domain and out-of-distribution is required a-priori. In this work we also make a comparison with Prior Networks trained via reverse KL-divergence (Malinin & Gales, 2019), where the Prior Networks (PN) are trained on the same datasets, both main and auxiliary, as the EnD^2(AUX) and EnD^2(AUX) model. Note, that for Ensemble Distribution Distillation, models can be distribution-distilled using any (potentially unlabeled) auxiliary data on which ensemble predictions can be obtained. Further note that in these experiments we explicitly chose to use the simpler VGG-16 architecture rather than more modern architectures like ResNet (He et al., 2016) as the goal of this work is to analyse the properties of Ensemble Distribution Distillation in a clean and simple configuration. The datasets and training configurations for all models are detailed in appendix A.

Table 3: Mean Classification Error, % PRR, test-set negative log-likelihood (NLL) and expected calibration error (ECE) on C10/C100/TIM across three models ±2σ.

<table>
<thead>
<tr>
<th>DSET</th>
<th>CRIT.</th>
<th>IND</th>
<th>ENSM</th>
<th>EnD</th>
<th>EnD^2</th>
<th>EnD^2(AUX)</th>
<th>EnD^2(AUX)</th>
<th>PN^2(AUX)</th>
</tr>
</thead>
<tbody>
<tr>
<td>C10</td>
<td>ERR</td>
<td>8.0 ±0.4</td>
<td>6.2 NA</td>
<td>6.7 ±0.3</td>
<td>7.3 ±0.2</td>
<td>6.7 ±0.2</td>
<td>6.9 ±0.2</td>
<td>7.5 ±0.3</td>
</tr>
<tr>
<td></td>
<td>PRR</td>
<td>84.6 ±1.2</td>
<td>86.8 NA</td>
<td>84.8 ±0.8</td>
<td>85.3 ±1.1</td>
<td>85.1 ±0.1</td>
<td>85.7 ±0.3</td>
<td>82.0 ±1.4</td>
</tr>
<tr>
<td></td>
<td>ECE</td>
<td>2.2 ±0.4</td>
<td>1.3 NA</td>
<td>2.6 ±0.2</td>
<td>1.0 ±0.2</td>
<td>2.6 ±0.6</td>
<td>2.2 ±0.4</td>
<td>12.0 ±0.7</td>
</tr>
<tr>
<td></td>
<td>NLL</td>
<td>0.25 ±0.01</td>
<td>0.19 NA</td>
<td>0.22 ±0.01</td>
<td>0.25 ±0.01</td>
<td>0.22 ±0.01</td>
<td>0.24 ±0.00</td>
<td>0.38 ±0.01</td>
</tr>
<tr>
<td>C100</td>
<td>ERR</td>
<td>30.4 ±0.3</td>
<td>26.3 NA</td>
<td>28.0 ±0.4</td>
<td>27.9 ±0.3</td>
<td>28.2 ±0.3</td>
<td>28.0 ±0.5</td>
<td>28.0 ±0.7</td>
</tr>
<tr>
<td></td>
<td>PRR</td>
<td>72.5 ±1.0</td>
<td>75.0 NA</td>
<td>73.1 ±0.5</td>
<td>73.7 ±0.7</td>
<td>74.0 ±0.3</td>
<td>74.0 ±0.2</td>
<td>63.7 ±0.8</td>
</tr>
<tr>
<td></td>
<td>ECE</td>
<td>9.3 ±0.8</td>
<td>1.2 NA</td>
<td>8.2 ±0.3</td>
<td>4.9 ±0.5</td>
<td>1.9 ±0.3</td>
<td>5.6 ±0.5</td>
<td>37.9 ±0.4</td>
</tr>
<tr>
<td></td>
<td>NLL</td>
<td>1.16 ±0.03</td>
<td>0.88 NA</td>
<td>1.06 ±0.01</td>
<td>1.14 ±0.01</td>
<td>0.98 ±0.00</td>
<td>1.14 ±0.01</td>
<td>1.87 ±0.03</td>
</tr>
<tr>
<td>TIM</td>
<td>ERR</td>
<td>41.8 ±0.6</td>
<td>36.6 NA</td>
<td>38.3 ±0.2</td>
<td>37.6 ±0.2</td>
<td>38.5 ±0.3</td>
<td>37.3 ±0.5</td>
<td>40.0 ±0.6</td>
</tr>
<tr>
<td></td>
<td>PRR</td>
<td>70.8 ±1.1</td>
<td>73.8 NA</td>
<td>72.2 ±0.2</td>
<td>73.1 ±0.1</td>
<td>72.6 ±1.3</td>
<td>72.7 ±1.1</td>
<td>62.3 ±0.6</td>
</tr>
<tr>
<td></td>
<td>ECE</td>
<td>18.3 ±0.8</td>
<td>3.8 NA</td>
<td>14.8 ±0.4</td>
<td>7.2 ±0.4</td>
<td>14.9 ±0.3</td>
<td>7.2 ±0.2</td>
<td>39.1 ±1.0</td>
</tr>
<tr>
<td></td>
<td>NLL</td>
<td>2.15 ±0.05</td>
<td>1.51 NA</td>
<td>1.77 ±0.01</td>
<td>1.83 ±0.02</td>
<td>1.78 ±0.01</td>
<td>1.84 ±0.02</td>
<td>2.61 ±0.01</td>
</tr>
</tbody>
</table>

1 The auxiliary training data for CIFAR-10 is CIFAR-100, and CIFAR-10 for CIFAR-100/TinyImageNet.
2 This is a consistent configuration to the EnD(AUX) and EnD^2(AUX) models, but constitutes a degraded OOD detection baseline for CIFAR-100 and TinyImageNet datasets due to the choice of auxiliary data.
Firstly, we investigate the ability of a single model to retain the ensemble’s classification and prediction-rejection (misclassification detection) performance after either Ensemble Distillation (EnD) or Ensemble Distribution Distillation (EnD\(^2\)), with results presented in Table 3. In terms of error rate and prediction rejection ratio (PRR), a higher PRR indicates that the model is able to better detect and reject incorrect predictions based on measures of uncertainty. This metric is detailed in Appendix B. Note that for prediction rejection we used the confidence of the max class, which also is a measure of total uncertainty, like entropy, but is more sensitive to the prediction (Malinin & Gales, 2018).

Table 3 shows that both EnD and EnD\(^2\) are able to retain both the improved classification and prediction-rejection performance of the ensemble relative to individual models trained with maximum likelihood on all datasets, both with and without auxiliary training data. Note, that on C100 and TIM, EnD\(^2\) yields marginally better classification performance than EnD. Furthermore, EnD\(^2\) also either consistently outperforms or matches EnD in terms of PRR on all datasets, both with and without auxiliary training data. This suggests that EnD\(^2\) is able to yield benefits on top of standard Ensemble Distillation due to retaining information about the diversity of the ensemble. In comparison, a Prior Network, while yielding a classification performance between that on an individual model and EnD\(_{\text{AUX}}\), performs consistently worse than all models in terms of PRR and for all datasets. This is likely because of the inappropriate choice of auxiliary training data and auxiliary loss weight (Malinin & Gales, 2019) for this task.

Secondly, it is known that ensembles yield improvements in the calibration of a model’s predictions (Lakshminarayanan et al., 2017; Ovadia et al., 2019). Thus, it is interesting to see whether EnD (Malinin & Gales, 2018) and EnD\(^2\) also is a measure of total uncertainty (CIFAR-10), which drives the model to be very under-confident. 

Ensemble Distribution Distillation seems to give similarly minor gains in NLL over a single model.

Table 4: OOD detection performance (mean % AUC-ROC ±2\(\sigma\)) for C10/C100/TIM models using measures of total (T.Unc) and knowledge (K.Unc) uncertainty.
Thirdly, Ensemble Distribution Distillation is investigated on the task of out-of-domain (OOD) input detection (Hendrycks & Gimpel, 2016), where measures of uncertainty are used to classify inputs as either in-domain (ID) or OOD. The ID examples are the test set of C10/100/TIM, and the test OOD examples are chosen to be the test sets of LSUN (Yu et al., 2015), C100 or TIM, such that the test OOD data is never seen by the model during training. Table 4 shows that the measures of uncertainty derived from the ensemble outperform those from a single neural network. Curiously, standard ensemble distillation (EnD) clearly fails to capture those gains on C10 and TIM, both with and without auxiliary training data, but does reach a comparable level of performance to the ensemble on C100. Curiously, EnD+aux performs worse than the individual models on C10. This is in contrast to results from (Englesson & Azizpour, 2019; Li & Hoiem, 2019), but the setups, datasets and evaluation considered there are very different to ours. On the other hand, Ensemble Distribution Distillation is generally able to reproduce the OOD detection performance of the ensemble. When auxiliary training data is used, EnD² is able to perform on par with the ensemble, indicating that it has successfully learned how the distribution of the ensemble behaves on unfamiliar data. These results suggest that not only is EnD² able to preserve information about the diversity of an ensemble, unlike standard EnD, but that this information is important to achieving good OOD detection performance. Notably, on the CIFAR-10 dataset PNs yield the best performance. However, due to generally inappropriate choice of OOD (auxiliary) training data, PNs show inferior (with one exception) OOD detection performance on CIFAR-100 and TIM.

Curiously, the ensemble sometimes displays better OOD detection performance using measures of total uncertainty. This is partly a property of the in-domain dataset - if it contains a small amount of data uncertainty, then OOD detection performance using total uncertainty and knowledge uncertainty should be almost the same (Malinin & Gales, 2018; Malinin, 2019). Unlike the toy dataset considered in the previous section, where significant data uncertainty was added, the image datasets considered here do have a low degree of data uncertainty. It is important to note that on more challenging tasks, which naturally exhibit a higher level of data uncertainty, we would expect that the decomposition would be more beneficial. Additionally, it is also possible that if different architectures, training regimes and ensembling techniques are considered, an ensemble with a better estimate of knowledge uncertainty can be obtained. Despite the behaviour of the ensemble, EnD², especially with auxiliary training data, tends to have better OOD detection performance using measures of knowledge uncertainty. This may be due to the use of a Dirichlet output distribution, which may not be able to fully capture the details of the ensemble’s behaviour. Furthermore, as EnD² is (implicitly) trained by minimizing the forward KL-divergence, which is zero-avoiding (Murphy, 2012), it is likely that the distribution learned by the model is ‘wider’ than the empirical distribution. This effect is explored in the next subsection and in appendix C.

5.1 Appropriateness of Dirichlet distribution

Throughout this work, a Prior Network that parametrizes a Dirichlet was used for distribution-distilling ensembles of models. However, the output distributions of an ensemble for the same input are not necessarily Dirichlet-distributed, especially in regions where the ensemble is diverse. In the previous section we saw that EnD² models tend to have higher NLL than EnD models, and while EnD² achieves good OOD detection performance, it doesn’t fully replicate the ensemble’s behaviour. Thus, in this section, we investigate how well a model which parameterizes a Dirichlet distribution is able to capture the exact behaviour of an ensemble of models, both in-domain and out-of-distribution.

Figure 4 shows histograms of total uncertainty, data uncertainty and total uncertainty yielded by an ensemble, EnD² and EnD² aux, models trained on the CIFAR-10 dataset. The top row shows the uncertainty histogram for ID data, and the bottom for test OOD data (a concatenation of LSUN and TIM). On in-domain data, EnD² is seemingly able to emulate the uncertainty metrics of the ensemble well, though does have a longer tail of high-uncertainty examples. This is expected, as on in-domain examples the ensemble will be highly concentrated around the mean. This behaviour can be adequately modelled by a Dirichlet. On the other hand, there is a noticeable mismatch between the ensemble and EnD² in the uncertainties they yield on OOD data. Here, EnD² consistently yields higher uncertainty predictions than the original ensemble. Notably, adding auxiliary training data makes the model yield even higher estimates of total uncertainty and data uncertainty. At the same time, by using auxiliary training data, the model’s distribution of knowledge uncertainty starts to look more like the ensembles, but shifted to the right (higher).
Altogether, this suggests that samples from the ensemble are diverse in a way that’s different from a Dirichlet distribution. Thus, as a consequence of this, combined with forward KL-divergence between the model and the empirical distribution of the ensemble being zero-avoiding, the model over-estimates the support of the empirical distribution, yielding an output which is both more diverse and higher entropy than the original ensemble. This does not seem to adversely impact OOD detection performance as measures of uncertainty for ID and OOD data are further spread apart and the rank ordering of ID and OOD data is either maintained or improved, which is supported by results from section 5. However, this does prevent the EnD$^2$ models from fully retaining the calibration quality of the ensemble. It is possible that the ensemble could be better modelled by a different output distribution, such as a mixture of Dirichlet distributions or a Logistic-normal distribution.

6 Conclusion

Ensemble Distillation approaches have become popular, as they allow a single model to achieve classification performance comparable to that of an ensemble at a lower computational cost. This work proposes the novel task Ensemble Distribution Distillation (EnD$^2$) — distilling an ensemble into a single model, such that it exhibits both the improved classification performance of the ensemble and retains information about its diversity. An approach to EnD$^2$ based on using Prior Network models is considered in this work. Experiments described in sections 4 and 5 show that on both artificial data and image classification tasks it is possible to distribution distill an ensemble into a single model such that it retains the classification performance of the ensemble. Furthermore, measures of uncertainty provided by EnD$^2$ models match the behaviour of an ensemble of models on artificial data, and EnD$^2$ models are able to differentiate between different types of uncertainty. However, this may require obtaining auxiliary training data on which the ensemble is more diverse in order to allow the distribution-distilled model to learn appropriate out-of-domain behaviour. On image classification tasks measures of uncertainty derived from EnD$^2$ models allow them to outperform both single NNs and EnD models on the tasks of misclassification and out-of-distribution input detection. These results are promising, and show that Ensemble Distribution Distillation enables a single model to capture more useful properties of an ensemble than standard Ensemble Distillation. Future work should further investigate properties of temperature annealing, investigate ways to enhance the diversity of an ensemble, consider different sources of ensembles and model architectures, and examine more flexible output distributions, such as mixtures of Dirichlets. Furthermore, while this work considered Ensemble Distribution Distillation only for classification problems, it can and should also be investigated for regression tasks. Finally, it may be interesting to explore combining Ensemble Distribution Distillation with standard Prior Network training.
REFERENCES


APPENDIX A  DATASETS, MODEL ARCHITECTURE AND TRAINING

Table 5: Description of datasets used in the experiments in terms of number of images and classes.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Train</th>
<th>Valid</th>
<th>Test</th>
<th>Classes</th>
</tr>
</thead>
<tbody>
<tr>
<td>CIFAR-10</td>
<td>50000</td>
<td>-</td>
<td>10000</td>
<td>10</td>
</tr>
<tr>
<td>CIFAR-100</td>
<td>50000</td>
<td>-</td>
<td>10000</td>
<td>100</td>
</tr>
<tr>
<td>TinyImagenet</td>
<td>100000</td>
<td>-</td>
<td>10000</td>
<td>200</td>
</tr>
<tr>
<td>LSUN (evaluation only)</td>
<td>-</td>
<td>-</td>
<td>10000</td>
<td>10</td>
</tr>
</tbody>
</table>

All models considered in this work were implemented in Pytorch (Paszke et al., 2017) using a variant of the VGG16 (Simonyan & Zisserman, 2015) architecture for image classification. DNN and EnD models were trained using the negative log-likelihood loss of the labels and the mean ensemble predictions respectively. EnD² models were trained using the negative log-likelihood of the ensemble’s output categorical distributions. All models were trained using the Adam (Kingma & Ba, 2015) optimizer, with a 1-cycle learning rate policy and dropout regularization. For all ensembles, models were trained using different random seed initialization, and using different seeds for shuffling the data. In addition, data augmentation was applied via random left-right flips, random shifts up to ±4 pixels and random rotations by up to ±15 degrees. Tables e details the training configurations for all models. Furthermore, batch normalisation was used for both Ensemble Distillation and Ensemble Distribution Distillation, but not for Prior Networks.

Table 6: Training Configurations. \( \eta_0 \) is the initial learning rate, \( T_0 \) is the initial temperature and 'Annealing' refers to whether a temperature annealing schedule was used. The batch size for all models was 128. Dropout rate is quoted in terms of probability of not dropping out a unit.

<table>
<thead>
<tr>
<th>Training Dataset</th>
<th>Model</th>
<th>( \eta_0 )</th>
<th>General</th>
<th>Distillation</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>DNN</td>
<td>45</td>
<td>30</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>EnD</td>
<td>90</td>
<td>60</td>
<td>0.7</td>
</tr>
<tr>
<td></td>
<td>EnD + AUX</td>
<td>( 10^{-3} )</td>
<td>90 60</td>
<td>0.7</td>
</tr>
<tr>
<td></td>
<td>EnD²</td>
<td>90</td>
<td>60</td>
<td>0.7</td>
</tr>
<tr>
<td></td>
<td>EnD² + AUX</td>
<td>90 60</td>
<td>0.7</td>
<td>10 Yes CIFAR-100</td>
</tr>
<tr>
<td></td>
<td>PN</td>
<td>5 ( \times 10^{-4} )</td>
<td>45 30</td>
<td>0.7</td>
</tr>
<tr>
<td>CIFAR-100</td>
<td>DNN</td>
<td>100</td>
<td>150</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>EnD</td>
<td>200</td>
<td>150</td>
<td>0.9</td>
</tr>
<tr>
<td></td>
<td>EnD + AUX</td>
<td>( 10^{-3} )</td>
<td>200 150</td>
<td>0.9</td>
</tr>
<tr>
<td></td>
<td>EnD²</td>
<td>200</td>
<td>150</td>
<td>0.9</td>
</tr>
<tr>
<td></td>
<td>EnD² + AUX</td>
<td>200 150</td>
<td>0.9</td>
<td>10 Yes CIFAR-10</td>
</tr>
<tr>
<td></td>
<td>PN</td>
<td>5 ( \times 10^{-4} )</td>
<td>100 70</td>
<td>0.7</td>
</tr>
<tr>
<td>TinyImageNet</td>
<td>DNN</td>
<td>10 ( \times 10^{-3} )</td>
<td>100 70</td>
<td>0.5</td>
</tr>
<tr>
<td></td>
<td>EnD</td>
<td>5 ( \times 10^{-4} )</td>
<td>200 150</td>
<td>0.8</td>
</tr>
<tr>
<td></td>
<td>EnD + AUX</td>
<td>( 5 \times 10^{-4} )</td>
<td>200 150</td>
<td>0.8</td>
</tr>
<tr>
<td></td>
<td>EnD²</td>
<td>5 ( \times 10^{-4} )</td>
<td>200 150</td>
<td>0.8</td>
</tr>
<tr>
<td></td>
<td>EnD² + AUX</td>
<td>( 5 \times 10^{-4} )</td>
<td>200 150</td>
<td>0.8</td>
</tr>
<tr>
<td></td>
<td>PN</td>
<td>5 ( \times 10^{-4} )</td>
<td>100 70</td>
<td>0.7</td>
</tr>
</tbody>
</table>

To create the transfer set \( D_{\text{ens}} \), ensembles were evaluated on the unaugmented CIFAR-10 and CIFAR-100 training examples. During distillation (both EnD and EnD²), models were trained on the augmented examples with the ensemble predictions on the corresponding unaugmented inputs.
A.1 Numerical issues with maximum likelihood Dirichlet training

As shown in equation\[8\] ensemble distribution distillation into a prior network via likelihood maximisation is equivalent to minimising the loss function:

\[
L(\phi, D_{\text{ens}}) = - \frac{1}{N} \sum_{i=1}^{N} \left[ \ln \Gamma(\hat{\alpha}_0^{(i)}) - \sum_{c=1}^{K} \ln \Gamma(\hat{\alpha}_c^{(i)}) + \frac{1}{M} \sum_{m=1}^{M} \sum_{i=1}^{K} (\hat{\alpha}_c^{(i)} - 1) \ln \pi_c^{(im)} \right]
\]  

(11)

If, due to numerical precision of the implementation, one of the ensemble member sample terms \(\pi_c^{(im)}\) gets rounded to 0.0, the log term \(\ln \pi_c^{(im)}\) in the above equation cannot be computed. To avoid this, we apply a small amount of central smoothing to the ensemble predictions:

\[
\pi_{c,\text{smoothed}}^{(im)} = (1 - \gamma)\pi_c^{(im)} + \gamma \frac{1}{K}
\]  

(12)

Where the smoothing parameter parameter \(\gamma\) has been set to \(1 \times 10^{-4}\) for all EnD\(^2\) experiments. Note that after applying central smoothing, each ensemble member’s predictions still sum up to 1.0.

A.2 Temperature Annealing Schedule

A fixed temperature of 2.5 was used for Ensemble Distillation as recommended in \cite{Hinton2015}, and was found to yield the best classification performance out of \{1, 2.5, 5, 10\}. Temperature annealing resulted in worse classification performance for Ensemble Distillation, and hence was not used in the experiments. For the temperature annealing schedule, the temperature was kept fixed to initial temperature \(T_0\) for the first half-cycle. For the second half-cycle we linearly decayed the initial temperature \(T_0\) down to 1.0. Then, the temperature was kept constant at 1.0 for the remainder of the epochs. For Ensemble Distribution Distillation, we found that an initial temperature of 10 performed best out of \{5, 10, 20\}. The choice of the particular annealing schedule used has not been tested extensively, and it is possible that other schedules that lead to faster training exist. However, we must point out that it was necessary to use temperature scaling to train models on the CIFAR-10, CIFAR-100 and TinyImageNet datasets.

Appendix B Assessing misclassification detection performance

In this work measures of uncertainty are used for two practical applications of uncertainty – misclassification detection and out-of-distribution sample detection. Both can be seen as an outlier detection task based on measures of uncertainty, where misclassifications are one form of outlier and out-of-distribution inputs are another form of outlier. These tasks can be formulated as threshold-based binary classification \cite{Hendrycks2016}. Here, a detector \(I_T(x)\) assigns the label 1 (uncertain prediction) if an uncertainty measure \(\mathcal{H}(x)\) is above a threshold \(T\), and label 0 (confident prediction) otherwise. This uncertainty measure can be any of the measures discussed in sections\[3\] and \[4\]

\[
I_T(x) = \begin{cases} 
1, & \mathcal{H}(x) > T \\
0, & \mathcal{H}(x) \leq T 
\end{cases}
\]  

(13)

Given a set of true positive examples \(D_p = \{x_p^{(i)}\}_{i=1}^{N_p}\) and a set of true negative examples \(D_n = \{x_n^{(j)}\}_{j=1}^{N_n}\) the performance of such a detection scheme can be evaluated at a particular threshold value \(T\) using the true positive rate \(t_p(T)\) and the false positive rate \(f_p(T)\):

\[
t_p(T) = \frac{1}{N_p} \sum_{i=1}^{N_p} I_T(x_p^{(i)}) \quad f_p(T) = \frac{1}{N_n} \sum_{j=1}^{N_n} I_T(x_n^{(j)})
\]  

(14)

The range of trade-offs between the true positive and the false positive rates can be visualized using a Receiver-Operating-Characteristic (ROC) and the quality of the possible trade-offs can be summarized using the area under the ROC curve (AUROC) \cite{Murphy2012}. If there are significantly more negatives than positives, however, this measure will over-estimate the performance of the model and yield a high AUROC value \cite{Murphy2012}. In this situation it is better to calculate the precision.
and recall of this detection scheme at every threshold value and plot them against each other on a Precision-Recall (PR) curve [Murphy, 2012]. The recall $R(T)$ is equal to the true positive rate $T_p(T)$, while precision measures the number of true positives among all samples labelled as positive:

$$
P(T) = \frac{\sum_{i=1}^{N_p} I_T(x_p^{(i)})}{\sum_{i=1}^{N_p} I_T(x_p^{(i)}) + \sum_{j=1}^{N_n} I_T(x_n^{(j)})}, \quad R(T) = \frac{1}{N_p} \sum_{i=1}^{N_p} I_T(x_p^{(i)})
$$

(15)

The quality of the trade-offs can again be summarized via the area under the PR curve (AUPR). For both the ROC and the PR curves an ideal detection scheme will achieve an AUC of 100%. A completely random detection scheme will have an AUROC of 50% and the AUPR will be the ratio of the number of positive examples to the total size of the dataset (positive and negative) [Murphy, 2012]. Thus, the recall is given by the error rate of the classifier. This makes it difficult to compare different models with different base error rates, as AUPR can increase both due to better misclassification detection and worse error rates.

Below we provide AUPR numbers for all models and datasets in order to illustrate how it can challenging to compare models using this metric. The difference in AUPR between models within a dataset is typically similar to the difference in classification performance. Thus, it is challenging to assesses whether a higher AUPR is due to better misclassification detection or higher error rate. Additionally, this table illustrate that confidence of the prediction is a better measure of uncertainty than entropy for this task. At the same, knowledge uncertainty yield much worse misclassification detection performance. These results are consistent with [Malinin & Gales, 2018, Malinin, 2019].

In this work we consider an alternative to using AUPR to assess misclassification detection performance proposed in [Malinin, 2019]. Consider the task of misclassification detection - ideally we would like to detect all of the inputs which the model has misclassified based on a measure of

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Model</th>
<th>Total Uncertainty Confidence</th>
<th>Uncertainty Entropy</th>
<th>Knowledge Uncertainty</th>
<th>% Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>C10</td>
<td>DNN</td>
<td>48.2 ± 2.7</td>
<td>47.0 ± 3.4</td>
<td>-</td>
<td>8.0 ± 0.4</td>
</tr>
<tr>
<td></td>
<td>ENS</td>
<td>43.9 ± NA</td>
<td>41.1 ± NA</td>
<td>36.8 ± NA</td>
<td>6.2 ± NA</td>
</tr>
<tr>
<td></td>
<td>EnD</td>
<td>44.6 ± 3.3</td>
<td>44.1 ± 3.9</td>
<td>37.6 ± 2.7</td>
<td>6.7 ± 0.3</td>
</tr>
<tr>
<td></td>
<td>EnD²</td>
<td>46.8 ± 1.2</td>
<td>46.1 ± 1.0</td>
<td>43.6 ± 0.9</td>
<td>7.3 ± 0.2</td>
</tr>
<tr>
<td></td>
<td>EnD_{AUX}</td>
<td>44.5 ± 2.0</td>
<td>43.8 ± 1.7</td>
<td>37.7 ± 1.1</td>
<td>6.7 ± 0.2</td>
</tr>
<tr>
<td></td>
<td>EnD²_{AUX}</td>
<td>46.5 ± 3.9</td>
<td>44.3 ± 3.5</td>
<td>39.4 ± 3.0</td>
<td>6.9 ± 0.2</td>
</tr>
<tr>
<td></td>
<td>PN-RKL</td>
<td>40.5 ± 3.0</td>
<td>38.5 ± 2.7</td>
<td>35.0 ± 2.3</td>
<td>7.5 ± 0.3</td>
</tr>
<tr>
<td>C100</td>
<td>DNN</td>
<td>69.8 ± 1.5</td>
<td>69.4 ± 1.4</td>
<td>-</td>
<td>30.4 ± 0.3</td>
</tr>
<tr>
<td></td>
<td>ENS</td>
<td>67.2 ± NA</td>
<td>64.0 ± NA</td>
<td>57.6 ± NA</td>
<td>26.3 ± NA</td>
</tr>
<tr>
<td></td>
<td>EnD</td>
<td>68.1 ± 0.8</td>
<td>67.6 ± 1.1</td>
<td>62.2 ± 1.8</td>
<td>28.0 ± 0.4</td>
</tr>
<tr>
<td></td>
<td>EnD²</td>
<td>68.3 ± 1.6</td>
<td>66.8 ± 1.5</td>
<td>63.4 ± 1.2</td>
<td>27.9 ± 0.3</td>
</tr>
<tr>
<td></td>
<td>EnD_{AUX}</td>
<td>69.3 ± 0.3</td>
<td>66.9 ± 0.2</td>
<td>58.0 ± 0.2</td>
<td>28.2 ± 0.3</td>
</tr>
<tr>
<td></td>
<td>EnD²_{AUX}</td>
<td>68.9 ± 0.4</td>
<td>66.7 ± 0.2</td>
<td>62.0 ± 1.0</td>
<td>28.0 ± 0.5</td>
</tr>
<tr>
<td></td>
<td>PN-RKL</td>
<td>60.1 ± 1.8</td>
<td>58.0 ± 1.8</td>
<td>52.8 ± 2.2</td>
<td>28.0 ± 0.7</td>
</tr>
<tr>
<td>TIM</td>
<td>DNN</td>
<td>77.9 ± 1.3</td>
<td>78.3 ± 1.2</td>
<td>-</td>
<td>41.8 ± 0.6</td>
</tr>
<tr>
<td></td>
<td>ENS</td>
<td>76.5 ± NA</td>
<td>74.8 ± NA</td>
<td>71.8 ± NA</td>
<td>36.6 ± NA</td>
</tr>
<tr>
<td></td>
<td>EnD</td>
<td>76.6 ± 0.5</td>
<td>77.3 ± 0.8</td>
<td>70.5 ± 1.1</td>
<td>38.3 ± 0.2</td>
</tr>
<tr>
<td></td>
<td>EnD²</td>
<td>77.0 ± 1.0</td>
<td>76.4 ± 0.9</td>
<td>74.8 ± 1.3</td>
<td>37.6 ± 0.2</td>
</tr>
<tr>
<td></td>
<td>EnD_{AUX}</td>
<td>76.9 ± 1.8</td>
<td>77.1 ± 2.1</td>
<td>69.9 ± 1.2</td>
<td>38.5 ± 0.3</td>
</tr>
<tr>
<td></td>
<td>EnD²_{AUX}</td>
<td>76.4 ± 1.7</td>
<td>75.8 ± 1.6</td>
<td>74.3 ± 2.1</td>
<td>37.3 ± 0.5</td>
</tr>
<tr>
<td></td>
<td>PN-RKL</td>
<td>73.0 ± 1.2</td>
<td>71.1 ± 1.1</td>
<td>60.7 ± 1.3</td>
<td>40.0 ± 0.6</td>
</tr>
</tbody>
</table>
uncertainty. Then, the model can either choose to not provide any prediction for these inputs, or they can be passed over or 'rejected' to an oracle (ie: human) to obtain the correct prediction. The latter process can be visualized using a rejection curve depicted in figure 5, where the predictions of the model are replaced with predictions provided by an oracle in some particular order based on estimates of uncertainty. If the estimates of uncertainty are ‘useless’, then, in expectation, the rejection curve would be a straight line from base error rate to the lower right corner. However, if the estimates of uncertainty are ‘perfect’ and always bigger for a misclassification than for a correct classification, then they would produce the ‘oracle’ rejection curve. The ‘oracle’ curve will go down linearly to 0% classification error at the percentage of rejected examples equal to the number of misclassifications. A rejection curve produced by estimates of uncertainty which are not perfect, but still informative, will sit between the ‘random’ and ‘oracle’ curves.

![Figure 5: Prediction Rejection Curves](image)

(a) Shaded area is $AR_{orc}$.

(b) Shaded area is $AR_{uns}$.

The quality of the rejection curve can be assessed by considering the ratio of the area between the ‘uncertainty’ and ‘random’ curves $AR_{uns}$ (orange in figure 5) and the area between the ‘oracle’ and ‘random’ curves $AR_{orc}$ (blue in figure 5). This yields the prediction rejection area ratio $PRR$:

$$PRR = \frac{AR_{uns}}{AR_{orc}}$$

(16)

A rejection area ratio of 1.0 indicates optimal rejection, a ratio of 0.0 indicates ‘random’ rejection. A negative rejection ratio indicates that the estimates of uncertainty are ‘perverse’ - they are higher for accurate predictions than for misclassifications. An important property of this performance metric is that it is independent of classification performance, unlike AUPR, and thus it is possible to compare models with different base error rates. Note, that similar approaches to assessing misclassification detection were considered in (Lakshminarayanan et al., 2017; Malinin et al., 2017).

**APPENDIX C**

**APPROPRIATENESS OF DIRICHLET DISTRIBUTION**

Section 5.1 explored the behaviour of measures of uncertainty predicted by models trained on the CIFAR-10 dataset. In this appendix we provide the same for models trained on CIFAR-100 and TinyImageNet. In general, the trends detailed in section 5.1 hold for these datasets as well - EnD consistently yield higher uncertainties than the original ensemble, likely as a consequence of the limitations of the Dirichlet distribution.

In addition, we have also provided histograms of total uncertainty for EnD and EnD$_{+AUX}$ models trained on CIFAR-10, CIFAR-100 and TinyImageNet relative to the ensemble for both in-domain and OOD data. Figure 7 shows that while all EnD models match the uncertainty of the ensemble on in-domain data for all datasets, EnD consistently under-estimates the uncertainty for OOD data. The only exception is EnD$_{+AUX}$ on CIFAR-100, where it matches the ensemble’s predictions well. This generally agrees with the results from section 5 where EnD$_{+AUX}$ is able to almost match the ensemble’s OOD detection performance.
In this paper we made several important design choices. Firstly, we chose to use rich ensembles of 100 models, in order to generate a good estimate of the empirical distribution of the ensemble on the simplex and assess whether the Dirichlet distribution was flexible enough to model it. Secondly, we used a temperature annealing schedule with an initial temperature of 10. In this section we conduct two ablation studies. Firstly, we assess whether Ensemble Distribution Distillation is possible when using an ensemble of 5, 20, 50 and 100 models. Secondly, we investigate a range of initial temperatures (1, 2, 5, 10, and 20) for temperature annealing for the models trained on the CIFAR-10 and CIFAR-100 datasets. Here, only the EnD$_{+\text{AUX}}$ model is considered. Results are presented in figures 8-13. All figures depict the mean across 3 models (for each configuration) ± 1 standard deviation.

Overall, the results show two important trends. Firstly, using using 20 models does better than using 5 models, but there are no conclusive gains for ensemble distribution distillation from using more than 20 models. This suggests that it is sufficient to use ensemble of fewer models for Ensemble Distribution Distillation with models which parameterize the Dirichlet Distribution. This is beneficial in terms computation and memory savings. It is possible, however, that if a more flexible distribution
Under review as a conference paper at ICLR 2020

Figure 7: Histograms of measures of total uncertainty derived from ensemble EnD, and EnD+ AUX on in-domain (ID) and test out-of-domain (OOD) data.

is used, such as a Mixture of Dirichlets, then it might be possible to derive further gains from a larger ensemble. The second trend is that it is necessary to use a temperature of at least 5 in order to successfully distribution-distill the ensemble. Using initial temperatures of 10 and 20 did not result in any significant further increase in performance. These results show that the temperature annealing process is important for Ensemble Distribution Distillation to work well.

Figure 8: CIFAR-10 Model Ablation
Figure 9: CIFAR-100 Model Ablation

Figure 10: CIFAR-10 and CIFAR-100 Model Ablation - Uncertainties
Figure 11: CIFAR-10 Temperature Ablation
Figure 12: CIFAR-100 Temperature Ablation

Figure 13: CIFAR-10 and CIFAR-100 Temperature Ablation - Uncertainties