ν-ENSEMBLES: IMPROVING DEEP ENSEMBLE CALIBRATION IN THE SMALL DATA REGIME

Anonymous authors
Paper under double-blind review

ABSTRACT

We present a method to improve the calibration of deep ensembles in the small data regime in the presence of unlabeled data. Our approach, which we name ν-ensembles, is extremely easy to implement: given an unlabeled set, for each unlabeled data point, we simply fit a different randomly selected label with each ensemble member. We provide a theoretical analysis based on a PAC-Bayes bound which guarantees that for such a labeling we obtain low negative log-likelihood and high ensemble diversity on testing samples. Empirically, through detailed experiments, we find that for low to moderately-sized training sets, ν-ensembles are more diverse and provide better calibration than standard ensembles, sometimes significantly.

1 INTRODUCTION

Deep ensembles have gained widespread popularity for enhancing both the testing accuracy and calibration of deep neural networks. This popularity largely stems from their ease of implementation and their consistent, robust improvements across various scenarios. Both empirically and theoretically, the performance of deep ensembles is intrinsically tied to their diversity (Fort et al., 2019; Masegosa, 2020). By averaging predictions from a more diverse set of models, we mitigate prediction bias and thereby enhance overall performance.

The conventional approach to introducing diversity within deep ensembles involves employing distinct random initializations for each ensemble member (Lakshminarayanan et al., 2017). As a result, these ensemble members converge towards different modes of the loss landscape, each corresponding to a unique predictive function. This baseline technique is quite difficult to surpass. Nevertheless, numerous efforts have been made to further improve deep ensembles by explicitly encouraging diversity in their predictions (Ramé & Cord, 2021; Yashima et al., 2022; Masegosa, 2020; Matteo et al., 2023).

These approaches typically encounter several challenges, which can be summarized as follows: *The improvements in test metrics tend to be modest, while the associated extra costs are substantial.* Firstly, diversity-promoting algorithms often involve considerably more intricate implementation details compared to randomized initializations. Secondly, the computational and memory demands of existing methods exceed those of the baseline by a significant margin. Additionally, some approaches necessitate extensive hyperparameter tuning, further compounding computational costs.

In light of these considerations, we introduce ν-ensembles, an algorithm designed to improve deep ensemble calibration and diversity with minimal deviations from the standard deep ensemble workflow. Moreover, our algorithm maintains the same computational and memory requirements as standard deep ensembles, resulting in linear increases in computational costs with the size of the unlabeled dataset.

Our contributions

- Given an ensemble of size K and an unlabeled set, we propose an algorithm that generates for each unlabeled data point K random labels without replacement and assigns from these a single random label to each ensemble member. For each ensemble member we then simply fit the training data (with its true labels) as well as the unlabeled data (with the

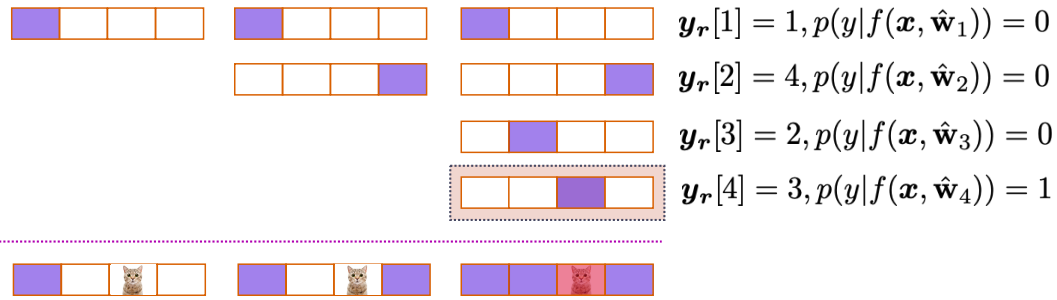


Figure 1: **Motivating ν-ensembles.** Consider a 4-class classification problem and an unlabeled sample $\boldsymbol{x}$ with true label $y = 3$. We sample $K = 4$ labels without replacement $\boldsymbol{y_r} = [1, 4, 2, 3]$ and fit them perfectly with ensemble members $\{\hat{\mathbf{w}}_1, \hat{\mathbf{w}}_2, \hat{\mathbf{w}}_3, \hat{\mathbf{w}}_4\}$. As we have sampled exhaustively all classes for this classification problem, exactly one of the sampled labels will be the correct one. The corresponding ensemble member $\hat{\mathbf{w}}_4$ will learn a useful feature from the input label pair $(\boldsymbol{x}, y)$. Noting that $p(y|\boldsymbol{x}, \hat{\mathbf{w}}_i)$ *is with respect to the true label* y, $p(y|\boldsymbol{x}, \hat{\mathbf{w}}_1) = 0, p(y|\boldsymbol{x}, \hat{\mathbf{w}}_2) = 0, p(y|\boldsymbol{x}, \hat{\mathbf{w}}_3) = 0, p(y|\boldsymbol{x}, \hat{\mathbf{w}}_4) = 1$ and the empirical variance will be $\hat{\mathbf{V}}(\hat{\rho}) = \frac{1}{2}\left[\frac{1}{K}\sum_j \left[(p(y|\boldsymbol{x}, \mathbf{w}_j) - \frac{1}{K}\sum_i (p(y|\boldsymbol{x}, \mathbf{w}_i)))^2\right]\right] = \frac{1}{2}\frac{4-1}{4} \cdot \frac{1}{4} = \frac{1}{2} \cdot \frac{3}{16}$ as computed in Proposition 1.

generated random labels). See Figure 1.

- We provide a PAC-Bayesian analysis of the test performance of the trained ensemble in terms of negative log-likelihood and diversity. On average, the final ensemble is guaranteed to be diverse, accurate, and well-calibrated on test data.

- We provide experiments for the in-distribution setting that demonstrate that for small to medium-sized training sets, ν-ensembles are better calibrated than standard ensembles in the most common calibration metrics.

- We also provide detailed experiments in the out-of-distribution setting and demonstrate that ν-ensembles remain significantly better calibrated than standard ensembles for a range of common distribution shifts.

2 Small to medium-sized training set setting

In the laboratory setting, deep learning models are typically trained and evaluated using large highly curated, and labeled datasets. However, real-world settings usually differ significantly. Labeled datasets are often small as the acquisition and labeling of new data is expensive, time-consuming, or simply not feasible. A small labeled training set is also often accompanied by a larger unlabeled set. A typical example where practitioners encounter such conditions is when applying deep learning in the medical field.

The small data regime has been explored in a number of works (Ratner et al., 2017; Balestriero et al., 2022; Zoph et al., 2020; Sorscher et al., 2022; Bornschein et al., 2020; Cubuk et al., 2020; Fabian et al., 2021; Zhao et al., 2019; Foong et al., 2021; Perez-Ortiz et al., 2021), both theoretical and practical. Two of the most common approaches for dealing with few training data, are using an ensemble of predictors, and/or using data augmentation to artificially create a larger training set.

We test our proposed ν-ensembles for a range of training set sizes, while applying data augmentation, and have found that we get performance gains for small to medium-sized training sets (1K - 10K samples). We emphasize that the "small data" regime is relative; more complex distributions require more data. As such ν-ensembles can be effective beyond these thresholds.

3 RELATED WORK ON IMPROVEMENTS OF DEEP ENSEMBLES

A number of approaches have been proposed to improve upon standard deep ensembles.

Diversity promoting objectives. Ramé & Cord (2021) propose to use a discriminator that forces the latent representations of each ensemble member just before the final classification layer to be diverse. They show consistent improvements for large-scale settings in terms of test accuracy and other metrics, however, their approach requires very extensive hyperparameter tuning. Yashima et al. (2022) encourage the latent representations just before the classification layer to be diverse by leveraging Stein Variational Gradient Descent (SVGD). They show improvements in robustness to non-adversarial noise. However, they do not show improvements over Ramé & Cord (2021) in other metrics.

Masegosa (2020); Ortega et al. (2022) propose optimizing a second-order PAC-Bayes bound to enforce diversity. In practice, this means estimating the mean likelihood of a true label across different ensemble members and "pushing" the different members to estimate a different value for their own likelihood. The authors show improvements for small-scale experiments, however, this comes at the cost of two gradient evaluations per data sample at each optimization iteration. The method closest to our approach is the very recently proposed Agree to Disagree algorithm (Matteo et al., 2023). Agree to disagree forces ensemble members to disagree with the other members on unlabeled data. Crucially, however, (and in contrast to our approach) the ensemble is constructed greedily, where a single new member is added at a time and is forced to disagree with the previous ones. The method is also evaluated only in the OOD setting.

The above methods exhibit all the shortcomings we previously described, where the cost of implementation, tuning and training cannot easily be justified: 1) the implementation differs significantly from standard ensembles (Ramé & Cord, 2021; Yashima et al., 2022; Masegosa, 2020; Matteo et al., 2023); 2) the computational complexity increases significantly (Ramé & Cord, 2021; Matteo et al., 2023); 3) and the algorithm requires extensive hyperparameter tuning (Ramé & Cord, 2021).

Bayesian approaches. One can also approach ensembles as performing approximate Bayesian inference (Wilson & Izmailov, 2020). Under this view, a number of approaches that perform approximate Bayesian inference can also be seen as constructing a deep ensemble (Izmailov et al., 2021; Wenzel et al., 2020a; Zhang et al., 2020; Immer et al., 2021; Daxberger et al., 2021). The samples from the approximate posterior that form the ensemble can be sampled locally around a single mode using the Laplace approximation (Immer et al., 2021; Daxberger et al., 2021) or from multiple modes using MCMC (Izmailov et al., 2021; Wenzel et al., 2020a; Zhang et al., 2020). While some approaches resort to stochastic MCMC approaches for computational efficiency (Wenzel et al., 2020a; Zhang et al., 2020), the authors of Izmailov et al. (2021) apply full-batch Hamiltonian Monte Carlo which is considered the gold standard in approximate Bayesian inference. D'Angelo & Fortuin (2021) propose a repulsive approach in terms of the neural network weights. They show that the resulting ensemble can be seen as Bayesian, however, they do not demonstrate consistent improvements across experimental setups.

One would hope that the regularizing effect of the Bayesian inference procedure would improve the resulting ensembles. Unfortunately, approximate Bayesian inference approaches are typically outperformed by *standard* deep ensembles (Ashukha et al., 2019). In particular, to achieve the same misclassification or negative log-likelihood error, MCMC approaches typically require many more ensemble members than standard ensembles.

Complementary works. Some works on diverse ensembles are compatible with our approach and can be used in conjunction with it.

Wenzel et al. (2020b) propose to induce diversity by training on different random initializations as well as different choices of hyperparameters such as the learning rate and the dropout rates in different layers. Ensemble members can be trained independently, and the approach results in consistent gains over standard ensembles. As we also train each ensemble member independently we could use hyperparameter ensembling to improve diversity. Jain et al. (2022) propose to create different training sets for each ensemble member using image transformations (for example edge detection filters) to bias different ensemble members towards different features. In a similar vein, Loh et al. (2023) encourage different ensemble members to be invariant or equivariant to different data trans-

formations. These approaches can also be used in conjunction with our method to further increase diversity.

Self-training. Jain et al. (2022) propose to pseudo-label unlabeled data using deep ensembles trained on labeled data. These pseudo-labeled data are then used to retrain the ensemble. This approach (known as self-training, see Lee et al., 2013) can improve significantly standard ensembles. We note however that it is complicated to implement and costly. First, unlabeled data have to be labeled in multiple rounds, a fraction at a time. Also, to be fully effective, ensembles have to be "distilled" into a final single network. Finally, care has to be taken that ensemble members capture diverse features. By contrast, our method requires a single random labeling of unlabeled data, followed by standard training and introduces a single hyperparameter that is easy to tune.

4 Diversity through unlabeled data

We now introduce some notation and then make precise our notions of train and test performance, as well as diversity.

We denote the learning sample $(X, Y) = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^n \in (\mathcal{X} \times \mathcal{Y})^n$, that contains n input-output pairs, and use the generic notation Z for an input-output pair (X, Y). Observations (X, Y) are assumed to be sampled randomly from a distribution $\mathcal{D}$. Thus, we denote $(X, Y) \sim \mathcal{D}^n$ the i.i.d observation of n elements. We consider loss functions $\ell : \mathcal{F} \times \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$, where $\mathcal{F}$ is a set of predictors $f : \mathcal{X} \to \mathcal{Y}$. We also denote the empirical risk $\hat{\mathcal{L}}^{\ell}_{X,Y}(f) = (1/n) \sum_i \ell(f, \boldsymbol{x}_i, y_i)$. We denote $\ell_{\text{nll}}(f, \boldsymbol{x}, y) = -\log(p(y|\boldsymbol{x}, f))$ the negative log-likelihood, where we assume that the outputs of f are normalized to form a probability distribution, and $p(y|\boldsymbol{x}, f)$ the probability of label y given $\boldsymbol{x}$ and f. Finally, let $\delta(x)$ be the Dirac delta function (in the following we suppress it's normalization where applicable).

Now let us assume that f is a deep neural network architecture, and $\hat{\rho}(\mathbf{w}) = \frac{1}{K} \sum_i \delta(\mathbf{w} = \hat{\mathbf{w}}_i)$ is a set of minima that form a deep ensemble. We are typically interested in minimizing $\mathbf{E}_{(y,\boldsymbol{x})\sim\mathcal{D}} \left[-\ln \frac{1}{K} \sum_i [p(y|\boldsymbol{x}, f(\boldsymbol{x}; \hat{\mathbf{w}}_i))] \right]$, the loss over new samples drawn from $\mathcal{D}$ for the ensemble predictor, that is: a predictor where we average the probabilities estimated per class by each ensemble member $\frac{1}{K} \sum_i p(y|\boldsymbol{x}, f(\boldsymbol{x}; \hat{\mathbf{w}}_i))$. The standard deep ensemble algorithm then simply minimizes $\forall i, \min_{\mathbf{w}_i} \hat{\mathcal{L}}^{\ell_{\text{nll}}}_Z(f(\boldsymbol{x}; \hat{\mathbf{w}}_i))$ for some training set Z.

Let us now assume that we have access not only to a training set Z but also to an unlabeled set U of size m. We can then present a PAC-Bayes bound* that links the loss on new test data to the loss on the training data as well as the diversity of the ensemble predictions on the unlabeled data.

Theorem 1. *With high probability over the training set Z and the unlabeled set U drawn from $\mathcal{D}$, for an ensemble $\hat{\rho}(\mathbf{w}) = \frac{1}{K} \sum_i \delta(\mathbf{w} = \hat{\mathbf{w}}_i)$ on $\mathcal{F}$ and all $\gamma \in (0, 2)$ simultaneously*

$$\mathbf{E}_{(y,\boldsymbol{x})\sim\mathcal{D}} \left[-\ln \frac{1}{K} \sum_i [p(y|\boldsymbol{x}, f(\boldsymbol{x}; \hat{\mathbf{w}}_i))] \right] \leq \frac{1}{K} \sum_i \left[\hat{\mathcal{L}}^{\ell_{\text{nll}}}_Z(f(\boldsymbol{x}; \hat{\mathbf{w}}_i)) \right] - \left(1 - \frac{\gamma}{2}\right) \hat{\mathbf{V}}(\hat{\rho}) + \frac{1}{K} \sum_i h\left(\|\hat{\mathbf{w}}_i\|_2^2\right), \quad (1)$$

where

$$\hat{\mathbf{V}}(\hat{\rho}) = \frac{1}{2m} \sum_U \left[\frac{1}{K} \sum_j \left[\left(p(y|\boldsymbol{x}, f(\boldsymbol{x}, \hat{\mathbf{w}}_j)) - \frac{1}{K} \sum_i p(y|\boldsymbol{x}, f(\boldsymbol{x}, \hat{\mathbf{w}}_i)) \right)^2 \right] \right] \quad (2)$$

is the empirical variance of the ensemble, and $h : \mathbb{R}^+ \to \mathbb{R}^+$ is a strictly increasing function.

The term $\frac{1}{K} \sum_i \left[\hat{\mathcal{L}}^{\ell_{\text{nll}}}_Z(f(\boldsymbol{x}; \hat{\mathbf{w}}_i)) \right]$ is simply the average negative log-likelihood of all the ensemble members on the training set Z. The term $\hat{\mathbf{V}}(\hat{\rho})$ captures our notion of diversity for the deep ensemble. Specifically, given a sample $(\boldsymbol{x}, y)$ it is the empirical variance of the likelihood $p(y|\boldsymbol{x}, f)$ of the

*Variants of this bound have appeared in recent works for majority vote classifiers (Thiemann et al., 2017; Wu & Seldin, 2022; Masegosa et al., 2020; Masegosa, 2020). However, to the best of our knowledge, this particular version is novel in the deep ensemble case.

Algorithm 1 ν-ensembles

Input: Weight of the unlabeled loss β, ℓ_2 regularization strength γ, training data Z, unlabeled data U, number of ensemble members K
Output: Ensemble $\mathcal{E}_K = \{\hat{\mathbf{w}}_1, \ldots, \hat{\mathbf{w}}_K\}$

for i in $\{1, \ldots, K\}$ **do**
  $U_i \leftarrow \{\}$
  for $\boldsymbol{x}$ in U **do**
    Sample y randomly without replacement from $[1, \ldots, c]$
    $U_i \leftarrow U_i \cup (\boldsymbol{x}, y)$
  end for
  $\hat{\mathbf{w}}_i \leftarrow$ Random Initialization
  $\min_{\hat{\mathbf{w}}_i} \hat{\mathcal{L}}_Z^{\ell_{\text{nll}}}(f(\boldsymbol{x}; \hat{\mathbf{w}}_i)) + \beta \hat{\mathcal{L}}_{U_i}^{\ell_{\text{nll}}}(f(\boldsymbol{x}; \hat{\mathbf{w}}_i)) + \gamma \|\hat{\mathbf{w}}_i\|_2^2$
end for

correct class y over all the ensemble members. The terms $h\left(\|\hat{\mathbf{w}}_i\|_2^2\right)$ capture a notion of complexity of the deep ensemble. If this term is too large, then it is possible that the ensemble has memorized the training and unlabeled sets leading to poor generalization on new data. From the above, we see that for a deep ensemble to generalize well to new data one needs to minimize its average training error, while maximizing its variance.

One could attempt to optimize the RHS of equation 1 directly by setting $U = Z$, through gradient descent. However, this introduces unnecessary complexity to the optimization objective, necessitates that all ensemble members are trained jointly, and also neglects potentially useful unlabeled data. We thus crucially evaluate the variance on a new unlabeled set U and not the training set Z. However, a careful reader would note that it is no longer possible to apply gradient descent directly to equation 1 as $\hat{\mathbf{V}}(\hat{\rho})$ depends on the unknown true label y. *We thus show in the following proposition that it is actually not necessary to know the true label y.* For each unlabeled sample $\boldsymbol{x}$, it simply suffices to draw K labels randomly without replacement and assign each of them to a different member of the deep ensemble. Then for $K = c$ exactly one of these labels will be the correct one. If each ensemble member fits these random labels perfectly then we can compute the variance term analytically for $K \leq c$.

Proposition 1. *Assume an unlabeled set $U \in \mathcal{D}^m$, c number of classes, and a labeling distribution $\mathcal{R}$ which for each sample $(\boldsymbol{x}, \cdot) \in U$ selects $K \leq c$ labels from $[1, \ldots, c]$ randomly without replacement such that $\boldsymbol{y_r} \in [1, \ldots, c]^K$. Let $\mathcal{A}$ be an algorithm that takes $\boldsymbol{y_r}$ as input and generates an ensemble $\hat{\rho}(\mathbf{w}) = \frac{1}{K}\sum_i \delta(\mathbf{w} = \hat{\mathbf{w}}_i)$ such that $\forall i, f(\boldsymbol{x}, \hat{\mathbf{w}}_i)$ perfectly fits $\boldsymbol{y_r}[i]$*

$$\mathbf{E}_{\hat{\rho}\sim\mathcal{A}}\left[\hat{\mathbf{V}}(\hat{\rho})\right] = \frac{K-1}{2cK} \tag{3}$$

where the randomness is over $\boldsymbol{y_r}$ and we suppress the index for the different unlabeled points.

Thus fitting $\boldsymbol{y_r} \sim \mathcal{R}$ guarantees in expectation through equation 3 a fixed level of variance, that strictly increases with the size of the ensemble. Taking the expectation on both sides of equation 1 we can also derive a high probability bound on $\mathbf{E}_{\hat{\rho}\sim\mathcal{A}}\mathbf{E}_{(y,\boldsymbol{x})\sim\mathcal{D}}\left[-\ln \frac{1}{K}\sum_i [p(y|\boldsymbol{x}, f(\boldsymbol{x}; \hat{\mathbf{w}}_i))]\right]$ given multiple samples from $\hat{\rho} \sim \mathcal{A}$, and subject to additional conditions on the training set and complexity terms (namely boundedness). We defer the technical details to the Appendix.

We thus propose algorithm 1 to train ν-ensembles. The proposed algorithm is extremely simple to implement. We simply need to construct K randomly labeled sets U_i, such that all the sets U_i contain different labels for all samples. We can then optimize

$$\hat{\mathcal{L}}_Z^{\ell_{\text{nll}}}(f(\boldsymbol{x}; \hat{\mathbf{w}}_i)) + \beta \hat{\mathcal{L}}_{U_i}^{\ell_{\text{nll}}}(f(\boldsymbol{x}; \hat{\mathbf{w}}_i)) + \gamma \|\hat{\mathbf{w}}_i\|_2^2 \tag{4}$$

with the optimization algorithm of our choice. In the above, β is the weight placed on the randomly labeled samples. Notably, doing hyperparameter optimization over β allows us to easily detect when ν-ensembles improve upon standard ensembles using a validation set, as for $\beta = 0$ we recover standard ensembles. The term $\gamma\|\hat{\mathbf{w}}_i\|_2^2$ results from equation 1, and coincides with standard weight decay regularization. Crucially we rely on being able to fit random labels. We note that it is well known that deep neural networks can fit random labels perfectly (Zhang et al., 2021).

5 IN-DISTRIBUTION AND OUT-OF-DISTRIBUTION EXPERIMENTS

We conducted two main types of experiments, evaluating (i) whether ν-ensembles improve upon standard ensembles for in-distribution testing data, (ii) whether the gains of ν-ensembles are robust to various distribution shifts.

To approximate the presence of unlabeled data using common classification datasets, given a training set Z, we reserve a validation set Z_{val}, and a smaller training set Z_{train} and use the remaining datapoints as a pool for unlabeled data U. We keep the testing data Z_{test} unchanged.

5.1 IN-DISTRIBUTION (ID) PERFORMANCE

To test in-distribution performance, we use the standard CIFAR-10 and CIFAR-100 datasets (Krizhevsky & Hinton, 2009). We explore a variety of dataset sizes. Specifically, for both datasets, we keep the original testing set such that $|Z_{\text{test}}| = 10000$, and we use 5000 samples from the training set as unlabeled data U and 5000 samples as validation data Z_{val}. For training, we use datasets Z_{train} of size $1000, 2000, 4000, 10000$ and 40000. We use three types of neural network architectures, a LeNet architecture LeCun et al. (1998), an MLP architecture with 2 hidden layers Goodfellow et al. (2016), and a WideResNet22 architecture Zagoruyko & Komodakis (2016). For both datasets, we used the standard augmentation setup of random flips + crops. We note that similar training-unlabeled set splits for CIFAR-10 and CIFAR-100 have been explored before in Alayrac et al. (2019); Jain et al. (2022).

We measure testing performance using accuracy as well as calibration on the testing set. Specifically, we measure calibration using the Expected Calibration Error (ECE) (Naeini et al., 2015), the Thresholded Adaptive Calibration Error (TACE) (Nixon et al., 2019), the Brier Score Reliability (Brier Rel.) (Murphy, 1973), and the Negative Log-Likelihood (NLL). We also measure the diversity of the ensemble on the test set using the average mutual information between ensemble member predictions. More specifically for each ensemble we treat its output as a random variable giving values in $[1, \dots, c]$. We compute the Mutual Information (MI) of this random variable between all ensemble pairs and take the average. Lower MI then corresponds to more diverse ensembles.

For both datasets, we first create an ensemble with $K = 10$ ensemble members and train each ensemble member using AdamW (Loshchilov & Hutter, 2017). For standard ensembles we simply minimize $\hat{\mathcal{L}}_Z^{\ell_{\text{nll}}}(f(\boldsymbol{x}; \hat{\mathbf{w}}_i)) + \gamma \|\hat{\mathbf{w}}_i\|_2^2$ for each ensemble member using different random initializations. For ν-ensembles we optimize equation 4. For hyperparameter tuning we perform a random search with 50 trials, using Hydra (Yadan, 2019). The details for the hyperparameter tuning ranges can be found in the Appendix. Table 1 presents the results for a training set of size 1000.

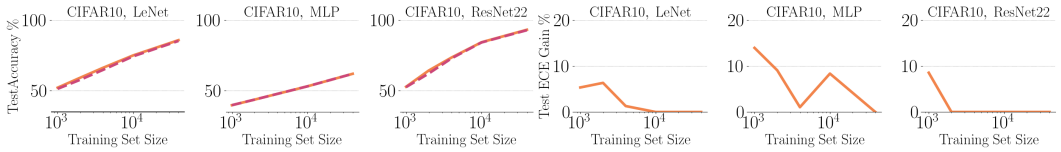
We see that ν-ensembles have comparable accuracy to standard ensembles but with significantly better calibration across all calibration metrics. We also see that ν-ensembles achieve significantly higher diversity between ensemble members. These results are consistent across all architectures for both CIFAR-10 and CIFAR-100. For the case of CIFAR-10, we see that the testing accuracy is low, however, this is to be expected due to the small size of the training dataset Z_{train}.

We also compare with Masegosa ensembles (Masegosa, 2020) and Agree to Disagree ensembles (Matteo et al., 2023) (we also attempted to implement DICE ensembles (Ramé & Cord, 2021) but could not replicate a version that converged consistently, despite correspondence with the authors). We see that both Masegosa and Agree to Disagree ensembles tend to underfit the data and have worse testing accuracy than ν-ensembles. In particular, Agree to Disagree ensembles also have in general worse calibration. Masegosa ensembles on the other hand have somewhat better calibration than ν-ensembles in most cases. Finally we also do temperature scaling for the Standard and ν-ensembles. The ν-ensembles + temperature scaling combination results in the best calibration. Our algorithm compares very favorably in terms of time and space complexity with both Masegosa and Agree to Disagree Ensembles. Standard and ν ensembles have $\mathcal{O}(1)$ memory cost as the ensemble size increases, if ensemble members are trained sequentially. On the other hand, Masegosa and Agree to Disagree ensembles in general scale like $\mathcal{O}(\mathrm{K})$ as all the ensemble members have to be trained jointly. Analyzing the computational cost is more complicated, however in general Masegosa ensembles require approximately $\times 2$ the computational time of Standard ensembles. Agree to Disagree ensembles scale roughly as $\mathcal{O}(\mathrm{K})$ as ensemble members have to be computed one at a time. In Figure 4 we compare the computational cost of Standard, ν and Agree to Disagree Ensembles.

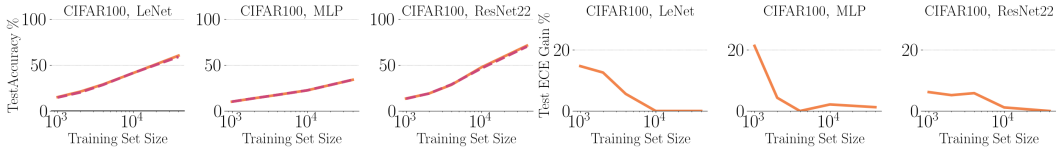
Table 1: **ID performance, 1000 training samples, 10 ensemble members.** ν-ensembles retain approximately the same accuracy as standard ensembles. At the same time, they achieve significantly better calibration in all calibration metrics. The improvements are consistent across all tested architectures and both datasets. We also observe that the Mutual Information (MI) of ν-ensembles is significantly lower than standard ensembles. Thus, ν-ensembles are more diverse than standard ensembles, which explains their improved calibration. These empirical observations are also consistent with our theoretical analysis. Masegosa and Agree to Disagree ensembles typically undefit and have lower testing accuracy than both Standard and ν-ensembles.

Dataset / Aug	Method	Acc ↑	ECE ↓	TACE ↓	Brier Rel. ↓	NLL ↓	MI ↓
CIFAR-10 / LeNet	Standard	**0.516**	0.176	0.034	0.133	2.043	1.320
	Agree Dis.	0.432	0.251	0.05	0.168	2.25	1.552
	Masegosa	0.492	0.103	0.024	**0.073**	1.454	**1.179**
	Tempering	0.517	0.024	0.0158	0.08	**1.419**	1.329
	ν-ensembles	0.506	0.133	0.028	0.118	1.664	1.201
	ν+Tempering	0.511	**0.014**	**0.0145**	0.08	1.437	1.215
CIFAR-10 / MLP	Standard	0.399	0.205	0.043	0.144	2.078	1.622
	Agree Dis.	0.354	0.358	0.066	0.239	3.201	1.547
	Masegosa	0.383	0.024	0.024	0.068	1.768	1.711
	Tempering	**0.402**	0.020	0.0134	**0.06**	1.71	1.625
	ν-ensembles	0.399	0.086	0.023	0.087	1.782	1.525
	ν+Tempering	0.401	**0.019**	**0.0133**	**0.06**	**1.69**	**1.554**
CIFAR-10 / ResNet22	Standard	**0.527**	0.087	0.024	0.106	1.690	0.939
	Agree Dis.	0.478	0.051	0.02	0.087	1.633	0.706
	Tempering	0.522	0.016	**0.017**	**0.086**	**1.354**	0.976
	ν-ensembles	0.527	0.014	0.017	0.082	1.436	0.675
	ν+Tempering	0.526	**0.010**	**0.017**	**0.086**	1.449	**0.691**
CIFAR-100 / LeNet	Standard	**0.149**	0.300	0.007	0.212	8.817	2.276
	Agree Dis.	0.113	0.229	0.007	0.156	7.568	1.628
	Masegosa	0.139	0.087	0.005	0.07	4.193	2.129
	Tempering	0.148	0.017	0.0039	0.049	**3.854**	2.236
	ν-ensembles	0.147	0.186	0.006	0.131	5.115	**1.826**
	ν+Tempering	0.144	**0.008**	**0.0038**	**0.048**	3.929	1.661
CIFAR-100 / MLP	Standard	0.101	0.183	0.007	0.114	5.173	3.142
	Agree Dis.	0.093	0.359	0.008	0.243	7.247	2.881
	Masegosa	0.093	0.257	0.008	0.16	6.134	3.103
	Tempering	0.102	**0.00822**	0.00417	**0.036**	4.155	3.128
	ν-ensembles	**0.103**	0.156	0.006	0.106	4.906	3.014
	ν+Tempering	0.103	0.019	**0.003**	**0.039**	**4.09**	**2.807**
CIFAR-100 / ResNet22	Standard	**0.137**	0.196	0.007	0.141	7.810	1.688
	Agree Dis.	0.132	0.172	0.007	0.124	6.831	1.708
	Tempering	0.136	**0.011**	0.004	0.040	**3.891**	1.608
	ν-ensembles	0.135	0.135	0.006	0.099	4.922	1.475
	ν+Tempering	0.131	0.018	**0.003**	**0.036**	3.930	**1.432**

We then explore the effect of increasing the dataset size. We plot the results of varying the training set size in $\{1000, 2000, 4000, 10000, 40000\}$ in Figure 2. We observe that ν-ensembles continue achieving the same accuracy as standard ensembles for all training set sizes. At the same time, they retain large improvements in calibration, in terms of the ECE, for small to medium size training sets. For larger training sets the improvements gradually decrease. Notably, there are differences between the easier CIFAR-10 and the more difficult CIFAR-100 dataset. Our calibration gains are significantly larger for the more difficult CIFAR-100 dataset. Furthermore, we retain these gains for larger training set sizes. In particular, we observe improvements for the ResNet22 architecture and 10000 training samples, while this is not the case for CIFAR-10.



(a) CIFAR-10



(b) CIFAR-100

Figure 2: **Varying the size of the training set.** For both standard and ν-ensembles, we vary the size of the training set Z_{train} to take values in $\{1000, 2000, 4000, 10000, 40000\}$. ν-ensembles have the same test accuracy as standard ensembles for all training set sizes. We also report the improvement in Expected Calibration Error (ECE) compared to standard ensembles. We see that, as the training size increases, the improvements decrease. Notably, we obtained larger improvements for the more difficult CIFAR-100 dataset than for the easier CIFAR-10 dataset. Also, we continue to have improvements for larger training set sizes. In particular, we observe improvements for the ResNet22 architecture at 10000 training samples while this is not the case for CIFAR-10.

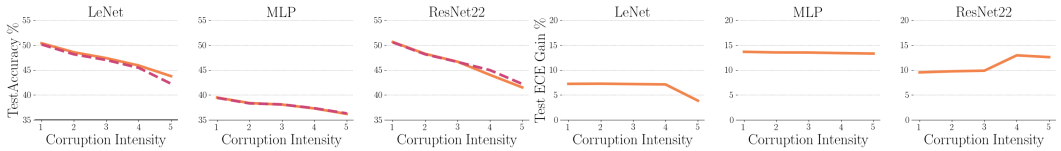


Figure 3: **CIFAR-10 robustness to common corruptions.** We apply 15 common image corruptions to the CIFAR-10 testing dataset for 5 levels of increasing intensity. For each intensity level, we then estimate the average testing accuracy and ECE across all corruption types, for both the standard ensemble and the ν-ensemble. We observe that the ν-ensemble retains approximately the same testing accuracy as the standard ensemble for all corruption levels. At the same time, the ν-ensemble is significantly better calibrated than the standard ensemble.

5.2 Out-of-distribution (OOD) generalization

We evaluated ν-ensembles and standard ensembles on difficult out-of-distribution tasks for the CIFAR-10 dataset, for the case of 1000 training samples. Specifically, we followed the approach introduced in Hendrycks & Dietterich (2018) which proposed to evaluate the robustness of image classification algorithms to 15 common corruption types. We apply the corruption in 5 levels of increasing severity and evaluate the average test accuracy and calibration in terms of ECE across all corruption types. We plot the results in Figure 3. We observe that ν-ensembles retain the same testing accuracy as standard ensembles. At the same time, they are significantly better calibrated in terms of the Expected Calibration Error. This holds for all tested architectures and for all corruption levels. We note that in the ResNet22 case, we see that ν-ensembles are particularly useful for high-intensity corruptions (the improvement in ECE increases from 10% to 15%).

5.3 Sampling with replacement

We are also interested in exploring how our method of sampling labels *without* replacement compares to sampling labels *with* replacement. Thus, we derive the following proposition.

Proposition 2. *Assume an unlabeled set $U \in \mathcal{D}^m$, c number of classes, and a labeling distribution $\mathcal{R}$ which for each sample $(\boldsymbol{x}, \cdot) \in U$ selects $K \leq c$ labels from $[1, \ldots, c]$ randomly **with** replacement such that $\boldsymbol{y_r} \in [1, \ldots, c]^K$. Let $\mathcal{A}$ be an algorithm that takes $\boldsymbol{y_r}$ as input and generates an ensemble*

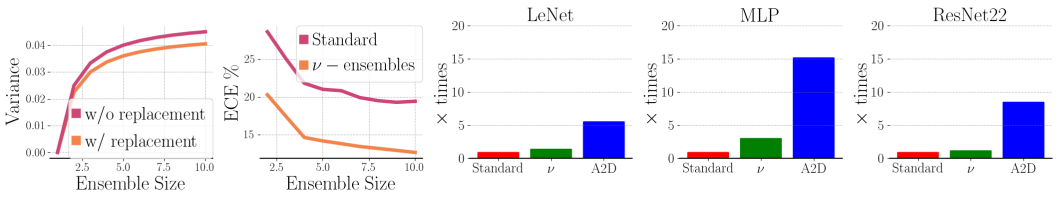


Figure 4: **ν-ensembles and other methods.** First: Let $c = 10$ and $K \in [1, \ldots, 10]$, we plot $\mathbf{E}_{\hat{\rho}\sim\mathcal{A}}[\hat{\mathbf{V}}(\hat{\rho})]$ with and without replacement. Sampling without replacement results in more diverse ensembles. Second: Improvements in ECE plateau around $K = 8$ for Standard ensembles, but continue improving for ν-ensembles. Other: we compare the training time of Standard, ν and Agree to Disagree ensembles, for the CIFAR-10 dataset with 1000 training samples and 5000 unlabeled samples. We plot $(\text{total training time})/(\text{epochs} * \text{ensemble size})$. Agree to Disagree ensembles have to be trained sequentially and have higher computational complexity for each member.

$\hat{\rho}(\mathbf{w}) = \frac{1}{K}\sum_i \delta(\mathbf{w} = \hat{\mathbf{w}}_i)$ *such that* $\forall i, f(\boldsymbol{x}, \hat{\mathbf{w}}_i)$ *perfectly fits* $\boldsymbol{y_r}[i]$

$$\mathbf{E}_{\hat{\rho}\sim\mathcal{A}}\left[\hat{\mathbf{V}}(\hat{\rho})\right] = \frac{1}{2}\left[\sum_r h(r)\binom{K}{r}\left(\frac{1}{c}\right)^r\left(1-\frac{1}{c}\right)^{K-r}\right] \tag{5}$$

where $h(r) = \frac{1}{K}\left[r\cdot\left(1-\frac{r}{K}\right)^2 + (K-r)\cdot\left(\frac{r}{K}\right)^2\right]$, *the randomness is over* $\boldsymbol{y_r}$ *and we suppress the index for the different unlabeled points.*

Comparing numerically propositions 1 and 2 in Figure 4, our theoretical analysis shows that for the same number of ensemble members, sampling without replacement results in higher variance and thus higher diversity and better calibration for our ensembles. We confirm our prediction by redoing the experiments in Table 1, but this time sampling with replacement. *On average, sampling without replacement results in better calibration across our different metrics.* Detailed results can be found in the Appendix.

6 Limitations

In our experiments, ν-ensembles demonstrate enhanced calibration performance when applied to standard ensembles, particularly in low to medium-data scenarios. However, in the context of a large data regime, we did not observe any notable improvements. Attempting to force the ensemble to learn random labels in such cases actually had a detrimental effect on calibration. This complex behaviour warrants a more nuanced theoretical analysis. The ability to predict in advance the specific training and unlabeled dataset sizes that would benefit from ν-ensembles would be a valuable asset. Additionally, it is worth noting that despite observing significant enhancements in calibration, counterintuitively we did not observe corresponding improvements in accuracy.

7 Conclusion

Deep ensembles have established themselves as a very strong baseline that is challenging to surpass. Not only do they consistently yield improvements across diverse settings, but they also do so with a very simple and efficient algorithm. Consequently, any algorithms aiming to enhance deep ensembles should prioritize efficiency and conceptual simplicity to ensure widespread adoption. In this work, we introduced ν-ensembles, a novel deep ensemble algorithm that achieves both goals. When presented with an unlabeled dataset, ν-ensembles generate distinct labelings for each ensemble member and subsequently fit both the training data and the randomly labeled data. Future directions of research include exploring the potential for ν-ensembles to outperform standard ensembles in the context of large datasets.

REFERENCES

Jean-Baptiste Alayrac, Jonathan Uesato, Po-Sen Huang, Alhussein Fawzi, Robert Stanforth, and Pushmeet Kohli. Are labels required for improving adversarial robustness? *Advances in Neural Information Processing Systems*, 32, 2019.

Arsenii Ashukha, Alexander Lyzhov, Dmitry Molchanov, and Dmitry Vetrov. Pitfalls of in-domain uncertainty estimation and ensembling in deep learning. In *International Conference on Learning Representations*, 2019.

Randall Balestriero, Ishan Misra, and Yann LeCun. A data-augmentation is worth a thousand samples: Analytical moments and sampling-free training. *Advances in Neural Information Processing Systems*, 35:19631–19644, 2022.

Jorg Bornschein, Francesco Visin, and Simon Osindero. Small data, big decisions: Model selection in the small-data regime. In *International conference on machine learning*, pp. 1035–1044. PMLR, 2020.

Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pp. 702–703, 2020.

Francesco D'Angelo and Vincent Fortuin. Repulsive deep ensembles are Bayesian. *Advances in Neural Information Processing Systems*, 34:3451–3465, 2021.

Erik Daxberger, Agustinus Kristiadi, Alexander Immer, Runa Eschenhagen, Matthias Bauer, and Philipp Hennig. Laplace Redux-Effortless Bayesian Deep Learning. *Advances in Neural Information Processing Systems*, 34, 2021.

Zalan Fabian, Reinhard Heckel, and Mahdi Soltanolkotabi. Data augmentation for deep learning based accelerated mri reconstruction with limited data. In *International Conference on Machine Learning*, pp. 3057–3067. PMLR, 2021.

Andrew Foong, Wessel Bruinsma, David Burt, and Richard Turner. How Tight Can PAC-Bayes be in the Small Data Regime? *Advances in Neural Information Processing Systems*, 34, 2021.

Stanislav Fort, Huiyi Hu, and Balaji Lakshminarayanan. Deep ensembles: A loss landscape perspective. *arXiv preprint arXiv:1912.02757*, 2019.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.

Dan Hendrycks and Thomas G Dietterich. Benchmarking neural network robustness to common corruptions and surface variations. *arXiv preprint arXiv:1807.01697*, 2018.

Alexander Immer, Maciej Korzepa, and Matthias Bauer. Improving predictions of Bayesian neural nets via local linearization. In *International Conference on Artificial Intelligence and Statistics*, pp. 703–711. PMLR, 2021.

Pavel Izmailov, Sharad Vikram, Matthew D Hoffman, and Andrew Gordon Gordon Wilson. What are Bayesian neural network posteriors really like? In *International Conference on Machine Learning*, pp. 4629–4640. PMLR, 2021.

Saachi Jain, Dimitris Tsipras, and Aleksander Madry. Combining diverse feature priors. In *International Conference on Machine Learning*, pp. 9802–9832. PMLR, 2022.

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

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in neural information processing systems*, 30, 2017.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Dong-Hyun Lee et al. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on challenges in representation learning, ICML*, volume 3, pp. 896. Atlanta, 2013.

Charlotte Loh, Seungwook Han, Shivchander Sudalairaj, Rumen Dangovski, Kai Xu, Florian Wenzel, Marin Soljacic, and Akash Srivastava. Multi-symmetry ensembles: Improving diversity and generalization via opposing symmetries. *arXiv preprint arXiv:2303.02484*, 2023.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

Andres Masegosa. Learning under model misspecification: Applications to variational and ensemble methods. *Advances in Neural Information Processing Systems*, 33:5479–5491, 2020.

Andrés Masegosa, Stephan Lorenzen, Christian Igel, and Yevgeny Seldin. Second order pac-bayesian bounds for the weighted majority vote. *Advances in Neural Information Processing Systems*, 33:5263–5273, 2020.

Pagliardini Matteo, Jaggi Martin, Fleuret François, and Karimireddy Sai Praneeth. Agree to disagree: Diversity through disagreement for better transferability. In *International Conference on Learning Representations*. ICLR, 2023.

Allan H Murphy. A new vector partition of the probability score. *Journal of Applied Meteorology and Climatology*, 12(4):595–600, 1973.

Mahdi Pakdaman Naeini, Gregory Cooper, and Milos Hauskrecht. Obtaining well calibrated probabilities using Bayesian binning. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015.

Jeremy Nixon, Michael W Dusenberry, Linchuan Zhang, Ghassen Jerfel, and Dustin Tran. Measuring calibration in deep learning. In *CVPR workshops*, volume 2, 2019.

Luis A Ortega, Rafael Cabañas, and Andres Masegosa. Diversity and generalization in neural network ensembles. In *International Conference on Artificial Intelligence and Statistics*, pp. 11720–11743. PMLR, 2022.

Maria Perez-Ortiz, Omar Rivasplata, Emilio Parrado-Hernandez, Benjamin Guedj, and John Shawe-Taylor. Progress in self-certified neural networks. *arXiv preprint arXiv:2111.07737*, 2021.

Alexandre Ramé and Matthieu Cord. DICE: Diversity in deep ensembles via conditional redundancy adversarial estimation. In *International Conference on Learning Representations*, 2021.

Alexander J Ratner, Henry Ehrenberg, Zeshan Hussain, Jared Dunnmon, and Christopher Ré. Learning to compose domain-specific transformations for data augmentation. *Advances in neural information processing systems*, 30, 2017.

Ben Sorscher, Robert Geirhos, Shashank Shekhar, Surya Ganguli, and Ari Morcos. Beyond neural scaling laws: beating power law scaling via data pruning. *Advances in Neural Information Processing Systems*, 35:19523–19536, 2022.

Niklas Thiemann, Christian Igel, Olivier Wintenberger, and Yevgeny Seldin. A strongly quasiconvex pac-bayesian bound. In *International Conference on Algorithmic Learning Theory*, pp. 466–492. PMLR, 2017.

Florian Wenzel, Kevin Roth, Bastiaan S Veeling, Jakub Swiatkowski, Linh Tran, Stephan Mandt, Jasper Snoek, Tim Salimans, Rodolphe Jenatton, and Sebastian Nowozin. How good is the Bayes posterior in deep neural networks really? *International Conference on Machine Learning*, 2020a.

Florian Wenzel, Jasper Snoek, Dustin Tran, and Rodolphe Jenatton. Hyperparameter ensembles for robustness and uncertainty quantification. *Advances in Neural Information Processing Systems*, 33:6514–6527, 2020b.

Andrew G Wilson and Pavel Izmailov. Bayesian deep learning and a probabilistic perspective of generalization. *Advances in neural information processing systems*, 33:4697–4708, 2020.

Yi-Shan Wu and Yevgeny Seldin. Split-kl and pac-bayes-split-kl inequalities for ternary random variables. *Advances in Neural Information Processing Systems*, 35:11369–11381, 2022.

Omry Yadan. Hydra - a framework for elegantly configuring complex applications. Github, 2019. URL https://github.com/facebookresearch/hydra.

Shingo Yashima, Teppei Suzuki, Kohta Ishikawa, Ikuro Sato, and Rei Kawakami. Feature space particle inference for neural network ensembles. In *International Conference on Machine Learning*, pp. 25452–25468. PMLR, 2022.

Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. In *British Machine Vision Conference 2016*. British Machine Vision Association, 2016.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning (still) requires rethinking generalization. *Communications of the ACM*, 64(3):107–115, 2021.

Ruqi Zhang, Chunyuan Li, Jianyi Zhang, Changyou Chen, and Andrew Gordon Wilson. Cyclical stochastic gradient MCMC for Bayesian deep learning. In *International Conference on Learning Representations*, 2020.

Amy Zhao, Guha Balakrishnan, Fredo Durand, John V Guttag, and Adrian V Dalca. Data augmentation using learned transformations for one-shot medical image segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 8543–8553, 2019.

Barret Zoph, Ekin D Cubuk, Golnaz Ghiasi, Tsung-Yi Lin, Jonathon Shlens, and Quoc V Le. Learning data augmentation strategies for object detection. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXVII 16*, pp. 566–583. Springer, 2020.