# $\nu$-ENSEMBLES: IMPROVING DEEP ENSEMBLE CALIBRATION IN THE SMALL DATA REGIME - APPENDIX

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

## A PROOFS

### A.1 PROOF OF THEOREM 1

**Theorem 1.** *(Theorem 2, Masegosa (2020)) For any distribution $\hat{\rho}$ on $\mathcal{F}$*

$$\mathbf{E}_{(y,\boldsymbol{x})\sim\mathcal{D}}\left[-\ln \mathbf{E}_{\mathbf{w}\sim\hat{\rho}}\left[p(y|\boldsymbol{x},f(\boldsymbol{x};\mathbf{w}))\right]\right] \leq \mathbf{E}_{\mathbf{w}\sim\hat{\rho}}\left[\mathcal{L}^{\ell_{\text{nll}}}_{(y,\boldsymbol{x})\sim\mathcal{D}}(f(\boldsymbol{x};\mathbf{w}))\right] - \mathbf{V}(\hat{\rho}) \tag{1}$$

*where $\mathbf{V}(\hat{\rho})$ is a variance term defined as*

$$\mathbf{V}(\hat{\rho}) = \mathbf{E}_{(y,\boldsymbol{x})\sim\mathcal{D}}\left[\frac{1}{2\max_{\mathbf{w}} p(y|\boldsymbol{x};\mathbf{w})}\mathbf{E}_{\mathbf{w}\sim\hat{\rho}}\left[\left(p(y|\boldsymbol{x},\mathbf{w}) - \mathbf{E}_{\mathbf{w}\sim\hat{\rho}}\left(p(y|\boldsymbol{x},\mathbf{w})\right)\right)^2\right]\right]. \tag{2}$$

We need to bound $\mathbf{V}(\hat{\rho})$ and $\mathbf{E}_{\mathbf{w}\sim\hat{\rho}}\left[\mathcal{L}^{\ell_{\text{nll}}}_{(y,\boldsymbol{x})\sim\mathcal{D}}(f(\boldsymbol{x};\mathbf{w}))\right]$ using their empirical versions. We will use a labeled training set $Z$ to bound the term $\mathbf{E}_{\mathbf{w}\sim\hat{\rho}}\left[\mathcal{L}^{\ell_{\text{nll}}}_{(y,\boldsymbol{x})\sim\mathcal{D}}(f(\boldsymbol{x};\mathbf{w}))\right]$ and an unlabeled set $U$ to bound $\mathbf{V}(\hat{\rho})$. To bound the terms we will use existing PAC-Bayes bounds. The variance term has to be rewritten in the form $\mathbf{E}_{\mathbf{w}\sim\hat{\rho}}\mathbf{E}_{(y,\boldsymbol{x})\sim\mathcal{D}}\left[L(y,\boldsymbol{x},\mathbf{w})\right]$ in which PAC-Bayes bounds are directly applicable.

Let us assume as in Masegosa (2020) that the model likelihood is bounded:

**Assumption 1.** *Masegosa (2020) There exists a constant $C < \infty$ such that $\forall \boldsymbol{x} \in \mathcal{X}$, $\max_{y,\mathbf{w}} p(y|\boldsymbol{x};\mathbf{w}) \leq C$.*

Note that this assumption holds for the classification setting with $C = 1$. Then the variance can be written as

$$\begin{aligned}
\mathbf{V}(\hat{\rho}) &= \frac{1}{2}\mathbf{E}_{(y,\boldsymbol{x})\sim\mathcal{D}}\left[\mathbf{E}_{\mathbf{w}\sim\hat{\rho}}\left[\left(p(y|\boldsymbol{x},\mathbf{w}) - \mathbf{E}_{\mathbf{w}\sim\hat{\rho}}\left(p(y|\boldsymbol{x},\mathbf{w})\right)\right)^2\right]\right] \\
&= \frac{1}{2}\mathbf{E}_{(y,\boldsymbol{x})\sim\mathcal{D}}\mathbf{E}_{\mathbf{w}\sim\hat{\rho}}\left[p(y|\boldsymbol{x},\mathbf{w})^2\right] - \frac{1}{2}\mathbf{E}_{(y,\boldsymbol{x})\sim\mathcal{D}}\left[\mathbf{E}_{\mathbf{w}\sim\hat{\rho}}p(y|\boldsymbol{x},\mathbf{w})\right]^2 \\
&= \frac{1}{2}\mathbf{E}_{(y,\boldsymbol{x})\sim\mathcal{D}}\mathbf{E}_{\mathbf{w}\sim\hat{\rho}}\left[p(y|\boldsymbol{x},\mathbf{w})^2\right] - \frac{1}{2}\mathbf{E}_{(y,\boldsymbol{x})\sim\mathcal{D}}\left[\mathbf{E}_{\mathbf{w}\sim\hat{\rho}}p(y|\boldsymbol{x},\mathbf{w})\mathbf{E}_{\mathbf{w}'\sim\hat{\rho}}p(y|\boldsymbol{x},\mathbf{w}')\right] \\
&= \frac{1}{2}\mathbf{E}_{(y,\boldsymbol{x})\sim\mathcal{D}}\mathbf{E}_{\hat{\rho}(\mathbf{w},\mathbf{w}')}\left[p(y|\boldsymbol{x},\mathbf{w})^2 - p(y|\boldsymbol{x},\mathbf{w})p(y|\boldsymbol{x},\mathbf{w}')\right] \\
&= \frac{1}{2}\mathbf{E}_{(y,\boldsymbol{x})\sim\mathcal{D}}\mathbf{E}_{\hat{\rho}(\mathbf{w},\mathbf{w}')}\left[L(y,\boldsymbol{x},\mathbf{w},\mathbf{w}')\right]
\end{aligned} \tag{3}$$

where $L(y,\boldsymbol{x},\mathbf{w},\mathbf{w}') = p(y|\boldsymbol{x},\mathbf{w})^2 - p(y|\boldsymbol{x},\mathbf{w})p(y|\boldsymbol{x},\mathbf{w}')$ and $\hat{\rho}(\mathbf{w},\mathbf{w}') = \hat{\rho}(\mathbf{w})\hat{\rho}(\mathbf{w}')$.

We can then use the following PAC-Bayes theorem to lower bound $\mathbf{V}(\hat{\rho})$ through it's empirical estimate, noting that $L(y,\boldsymbol{x},\mathbf{w},\mathbf{w}') \leq 1$ which is a requirement for this bound.

**Theorem 2.** *(PAC-Bayes-$\lambda$, Thiemann et al. (2017)). For any probability distribution $\pi$ on $\mathcal{F}$ that is independent of $U$ and any $\delta_1 \in (0,1)$, with probability at least $1-\delta_1$ over a random draw of a sample $U$, for all distributions $\hat{\rho}$ on $\mathcal{F}$ and all $\gamma \in (0,2)$ simultaneously and a bounded loss $L \leq 1$*

$$\mathbf{E}_{\mathbf{w}\sim\hat{\rho}}\mathbf{E}_{(y,\boldsymbol{x})\sim\mathcal{D}}\left[L(y,\boldsymbol{x},\mathbf{w})\right] \geq \left(1-\frac{\gamma}{2}\right)\mathbf{E}_{\mathbf{w}\sim\hat{\rho}}\frac{1}{m}\sum_{(y,\boldsymbol{x})\in U}\left[L(y,\boldsymbol{x},\mathbf{w})\right] - \frac{\mathrm{KL}(\hat{\rho}||\pi) + \ln(2\sqrt{m}/\delta)}{\gamma m} \tag{4}$$

We then turn to the term $\mathbf{E}_{\mathbf{w}\sim\hat{\rho}}\left[\mathcal{L}^{\ell_{\text{nll}}}_{(y,\boldsymbol{x})\sim\mathcal{D}}(f(\boldsymbol{x};\mathbf{w}))\right]$ where $L$ is unbounded due to the NLL loss. We will use the following bound:

**Theorem 3.** *( Alquier et al. (2016)). For any probability distribution $\pi$ on $\mathcal{F}$ that is independent of $Z$ and any $\delta_2 \in (0,1)$, with probability at least $1-\delta_2$ over a random draw of a sample $Z$, for all distributions $\hat{\rho}$ on $\mathcal{F}$ and $\gamma > 0$*

$$\mathbf{E}_{\mathbf{w}\sim\hat{\rho}}\left[\mathcal{L}^{\ell_{\text{nll}}}_{(y,\boldsymbol{x})\sim\mathcal{D}}(f(\boldsymbol{x};\mathbf{w}))\right] \leq \mathbf{E}_{\mathbf{w}\sim\hat{\rho}}\left[\hat{\mathcal{L}}^{\ell_{\text{nll}}}_{Z}(f(\boldsymbol{x};\mathbf{w}))\right] + \frac{\text{KL}(\hat{\rho}||\pi) + \ln(\frac{1}{\delta}) + \psi_{\pi,\mathcal{D}}(\gamma,n)}{\gamma n} \tag{5}$$

*where*

$$\psi_{\pi,\mathcal{D}}(\gamma,n) = \ln \mathbf{E}_{\pi}\mathbf{E}_{\mathcal{D}}\left[e^{\gamma n\left(\mathcal{L}^{\ell_{\text{nll}}}_{(y,\boldsymbol{x})\sim\mathcal{D}}(f(\boldsymbol{x};\mathbf{w}))-\hat{\mathcal{L}}^{\ell_{\text{nll}}}_{Z}(f(\boldsymbol{x};\mathbf{w}))\right)}\right]. \tag{6}$$

By setting $\gamma_1 = \gamma_2 = \gamma/2$ and taking a union bound we then get:

**Theorem 4.** *For any probability distribution $\pi$ on $\mathcal{F}$ that is independent of $U$ and $Z$ and any $\delta \in (0,1)$, with probability at least $1-\delta$ over a random draw of a sample $U$ and $Z$, for all distributions $\hat{\rho}$ on $\mathcal{F}$ and all $\gamma \in (0,2)$ simultaneously*

$$\begin{aligned}
&\mathbf{E}_{(y,\boldsymbol{x})\sim\mathcal{D}}\left[-\ln \mathbf{E}_{\mathbf{w}\sim\hat{\rho}}\left[p(y|\boldsymbol{x},f(\boldsymbol{x};\mathbf{w}))\right]\right] \leq \\
&\mathbf{E}_{\mathbf{w}\sim\hat{\rho}}\left[\hat{\mathcal{L}}^{\ell_{\text{nll}}}_{Z}(f(\boldsymbol{x};\mathbf{w}))\right] + \frac{\text{KL}(\hat{\rho}||\pi) + \ln(1/\delta) + \psi_{\pi,\mathcal{D}}(\gamma,n)}{\gamma n} \\
&- \left(1-\frac{\gamma}{2}\right)\hat{\mathbf{V}}(\hat{\rho}) + \frac{\text{KL}(\hat{\rho}||\pi) + \ln(2\sqrt{m}/\delta)}{\gamma m}.
\end{aligned} \tag{7}$$

What remains is to define the prior $\pi$ and posterior $\hat{\rho}$ distributions appropriately. We first set $\hat{\rho}(\mathbf{w}) = \frac{1}{K}\sum_i \delta(\mathbf{w} = \hat{\mathbf{w}}_i)$ which denotes an ensemble. We then follow Masegosa (2020) in properly defining the KL between $\hat{\rho}(\mathbf{w})$ and a given prior. Specifically, we restrict ourselves to a new family of priors, denoted $\pi_F(\mathbf{w})$. For any prior $\pi_F(\mathbf{w})$ within this family, its support is contained in $\mathbf{w}_F$, which denotes the space of real number vectors of dimension M that can be represented under a finite-precision scheme using F bits to encode each element of the vector. So we have $supp(\pi_F) \subseteq \mathbf{w}_F \subseteq \mathcal{R}^M$. This prior distribution $\pi_F$ can be expressed as, $\pi_F(\mathbf{w}) = \sum_{\mathbf{w}'\in\mathbf{w}_F} w_{\mathbf{w}'}\delta(\mathbf{w} = \mathbf{w}')$ where $w_{\mathbf{w}'}$ are positive scalar values parametrizing this prior distribution. They satisfy $w_{\mathbf{w}'} \geq 0$ and $\sum w_{\mathbf{w}'} = 1$. In this way, we can define a finite-precision counterpart to the Gaussian distribution where $w_{\mathbf{w}'} = \frac{1}{A}e^{-||\mathbf{w}'||_2^2}$ and $A$ is an appropriate normalization constant.

Puting everything back in equation 7 we get

$$\begin{aligned}
&\mathbf{E}_{(y,\boldsymbol{x})\sim\mathcal{D}}\left[-\ln\frac{1}{K}\sum_i\left[p(y|\boldsymbol{x},f(\boldsymbol{x};\hat{\mathbf{w}}_i))\right]\right] \\
&\qquad\leq \frac{1}{K}\sum_i\left[\hat{\mathcal{L}}^{\ell_{\text{nll}}}_{Z}(f(\boldsymbol{x};\hat{\mathbf{w}}_i))\right] - \left(1-\frac{\gamma}{2}\right)\hat{\mathbf{V}}(\hat{\rho}) + \frac{1}{K}\sum_i h\left(\|\hat{\mathbf{w}}_i\|_2^2\right),
\end{aligned} \tag{8}$$

where

$$h\left(\|\hat{\mathbf{w}}_i\|_2^2\right) = \frac{\|\hat{\mathbf{w}}_i\|_2^2 + \ln A + K\ln(1/\delta) + K\psi_{\pi,\mathcal{D}}(\gamma,n)}{\gamma n} + \frac{\|\hat{\mathbf{w}}_i\|_2^2 + \ln A + K\ln(2\sqrt{m}/\delta)}{\gamma m}, \tag{9}$$

and which holds for any $\delta \in (0,1)$, with probability at least $1-\delta$ over a random draw of a sample $U$ and $Z$.

Some further technical points need to be discussed at this point. Formally, Theorem 3 holds for a single value of $\gamma$. In order to combine both PAC-Bayes bounds we would need to form a grid over $\gamma$ in the range $(0,2)$ and do a union bound over this grid. The combined bound would then hold only for values on this grid. This results analysis only results in a negligible loosening of the bound (Dziugaite & Roy, 2017) and as such we neglect this discussion.

Since we have defined our bound in the discrete setting we cannot technically take derivatives of the resulting objective. However, as discussed in Masegosa (2020) during optimization we simply use the continuous version of all functions, knowing that we will arrive withing a solution of finite precision.

## B ADDITIONAL CONDITIONS FOR A HIGH-PROBABILITY BOUND

Given inequality 8, we can take the expectation over the proposed algorithm, $\hat{\rho} \sim \mathcal{A}$, to obtain

$$\mathbf{E}_{\hat{\rho}\sim\mathcal{A}}\mathbf{E}_{(y,\boldsymbol{x})\sim\mathcal{D}}\left[-\ln\frac{1}{K}\sum_i\left[p(y|\boldsymbol{x},f(\boldsymbol{x};\hat{\mathbf{w}}_i))\right]\right]$$
$$\leq \mathbf{E}_{\hat{\rho}\sim\mathcal{A}}\left[\frac{1}{K}\sum_i\left[\hat{\mathcal{L}}_Z^{\ell_{\text{nll}}}(f(\boldsymbol{x};\hat{\mathbf{w}}_i))\right]\right] - \left(1-\frac{\gamma}{2}\right)\frac{K-1}{2cK} + \mathbf{E}_{\hat{\rho}\sim\mathcal{A}}\left[\frac{1}{K}\sum_i h\left(\|\hat{\mathbf{w}}_i\|_2^2\right)\right], \quad (10)$$

which holds for any $\delta \in (0,1)$, with probability at least $1-\delta$ over a random draw of a sample $U$ and $Z$.

Then, setting $L_1(\hat{\rho}) = \frac{1}{K}\sum_i\left[\hat{\mathcal{L}}_Z^{\ell_{\text{nll}}}(f(\boldsymbol{x};\hat{\mathbf{w}}_i))\right]$ and $L_2(\hat{\rho}) = \frac{1}{K}\sum_i h\left(\|\hat{\mathbf{w}}_i\|_2^2\right)$ we note that both $L_1$ and $L_2$ are in general unbounded. To obtain a high-probability bound on $\mathbf{E}_{\hat{\rho}\sim\mathcal{A}}\mathbf{E}_{(y,\boldsymbol{x})\sim\mathcal{D}}\left[-\ln\frac{1}{K}\sum_i\left[p(y|\boldsymbol{x},f(\boldsymbol{x};\hat{\mathbf{w}}_i))\right]\right]$ we need additional conditions on $\mathcal{A}$ namely that it outputs $\hat{\rho}$ such that $L_1(\hat{\rho}) \leq B$ and $L_2(\hat{\rho}) \leq C$ where $B, C$ are positive constants.

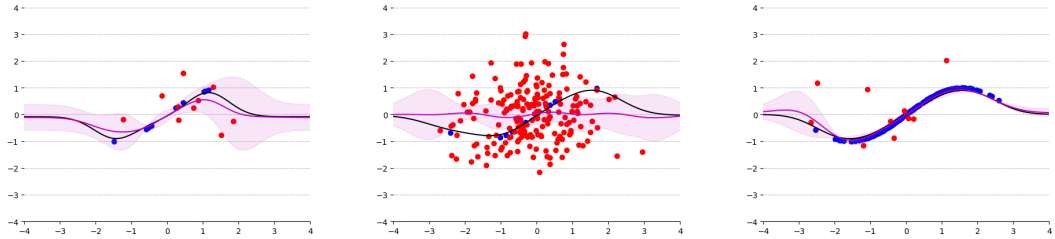
Then, for a finite sample $R \in \mathcal{A}^r$ and using Hoeffding's inequality and applying a union bound we can write

$$\mathbf{E}_{\hat{\rho}\sim\mathcal{A}}\mathbf{E}_{(y,\boldsymbol{x})\sim\mathcal{D}}\left[-\ln\frac{1}{K}\sum_{i\in\hat{\rho}}\left[p(y|\boldsymbol{x},f(\boldsymbol{x};\hat{\mathbf{w}}_i))\right]\right]$$
$$\leq \frac{1}{r}\sum_{\hat{\rho}\in R}\left[\frac{1}{K}\sum_{i\in\hat{\rho}}\left[\hat{\mathcal{L}}_Z^{\ell_{\text{nll}}}(f(\boldsymbol{x};\hat{\mathbf{w}}_i))\right]\right] + \sqrt{\frac{B^2\ln 1/b}{2r}}$$
$$- \left(1-\frac{\gamma}{2}\right)\frac{K-1}{2cK} + \frac{1}{r}\sum_{\hat{\rho}\in R}\left[\frac{1}{K}\sum_{i\in\hat{\rho}} h\left(\|\hat{\mathbf{w}}_i\|_2^2\right)\right] + \sqrt{\frac{C^2\ln 1/c}{2r}},$$

which holds with probability $1-(\delta+b+c)$ over the random draws of $U \in \mathcal{D}^m$, $Z \in \mathcal{D}^n$ and $R \in \mathcal{A}^r$ for $b, c \in (0,1)$. The bound still holds *for the expectation* over $\hat{\rho} \sim \mathcal{A}$ and not with high probability for a single draw from $\mathcal{A}$. It guarantees that on average, ensembles that fit the training data and the randomly labeled data well, while having low complexity will generalize well to unseen data. In our experimental section, however, we have found that optimizing a single ensemble using our $\nu$-ensemble objective achieves all the desirable properties.

## C TOY EXAMPLE

We investigate using a toy example the effect of varying the number of training and unlabeled samples. We first generate labeled data by setting $y = \sin(x)$. We then create multiple pseudo-labeled datasets by sampling $\boldsymbol{x} \sim \mathcal{N}(0, \mathbf{I})$ once and then sampling $y_{\boldsymbol{x}} \sim \mathcal{N}(0, \mathbf{I})\forall x$, K times where K is the ensemble size. We compare two predictors. One is a single SVM that fits only the training data (black curve). The second is an ensemble of K SVMs where each regressor fits the training data together with one of the K unlabeled datasets (purple curve). We fit the data using a support vector regressor with a gaussian kernel and the default hyperparameter values of scikit-learn Buitinck et al. (2013). In 1a when the number of training samples is small and the number of unlabeled samples is not too large, our ensemble learns to both fit the training data as well as have high uncertainty



(a) $|Z_{\text{train}}| = 10, |Z_{\text{unlabeled}}| = 10$ (b) $|Z_{\text{train}}| = 10, |Z_{\text{unlabeled}}| = 200$ (c) $|Z_{\text{train}}| = 200, |Z_{\text{unlabeled}}| = 10$

Figure 1: Left: the ensemble fits the training data and also has high uncertainty out-of-sample. Middle: the ensemble underfits the training dataset. Right: the size of the training dataset is large making the unlabeled data redundant or even detrimental to performance.

out-of-sample. In 1b when the unlabeled dataset becomes too large it also covers the space that the training datapoints cover. We see that then the ensemble underfits the data. In 1c when the size of the training dataset is large we do not need unlabeled data (and we might even hurt performance if we force our ensemble to fit an unlabeled dataset).

## D SAMPLING WITHOUT REPLACEMENT

**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 $[1, \dots, c]$ randomly **without** replacement such that $\boldsymbol{y_r} \in [1, \dots, c]^K$. 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 $\boldsymbol{y_r}[i]$*

$$\mathbf{E}_{\hat{\rho}\sim\mathcal{A}}\left[\hat{\mathbf{V}}(\hat{\rho})\right] = \frac{K-1}{2cK} \tag{11}$$

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

*Proof.* We first discuss some preliminaries. We assume that each ensemble member fits the label assigned to it perfectly. Given a sample $(\boldsymbol{x}, y)$ and $K$ randomly sampled labels $\boldsymbol{y_r} \in [1, \dots, c]^K$, without replacement, where only the $a$th label is the true label $y$, we have $p(y|\boldsymbol{x}, \mathbf{w}_a) = 1$ and $p(y|\boldsymbol{x}, \mathbf{w}_i) = 0,\ \forall i \neq a$.

The expectation of the variance term can now be simply obtained by separating the cases when $y$ is in the random labels $\boldsymbol{y_r} \in [1, \dots, c]^K$ and the cases when it is not. We get

$$\begin{aligned}
\mathbf{E}_{\hat{\rho}\sim\mathcal{A}}\left[\hat{\mathbf{V}}(\hat{\rho})\right] &= \mathbf{E}_{\hat{\rho}\sim\mathcal{A}}\left[\frac{1}{2m}\sum_{(\boldsymbol{x},y)\in U}\left[\frac{1}{K}\sum_j\left[\left(p(y|\boldsymbol{x},\mathbf{w}_j) - \frac{1}{K}\sum_i\left(p(y|\boldsymbol{x},\mathbf{w}_i)\right)\right)^2\right]\right]\right] \\
&= \frac{1}{2m}\sum_U\left[\frac{1}{K}\sum_j\left[\left(p(y|\boldsymbol{x},\mathbf{w}_j) - \frac{1}{K}\sum_i\left(p(y|\boldsymbol{x},\mathbf{w}_i)\right)\right)^2\right]\cdot\int\mathbb{I}\{y \text{ in randomized labels}\}dr\right. \\
&\quad \left. + \frac{1}{K}\sum_j\left[\left(p(y|\boldsymbol{x},\mathbf{w}_j) - \frac{1}{K}\sum_i\left(p(y|\boldsymbol{x},\mathbf{w}_i)\right)\right)^2\right]\cdot\int\mathbb{I}\{y \text{ not in randomized labels}\}dr\right] \\
&= \frac{1}{2m}\sum_U\left[\frac{K-1}{K^2}\cdot\int\mathbb{I}\{y \text{ in randomized labels}\}dr + 0\cdot\int\mathbb{I}\{y \text{ not in randomized labels}\}dr\right] \\
&= \frac{1}{2m}\sum_U\frac{K-1}{K^2}\cdot\frac{K}{c} \\
&= \frac{K-1}{2cK}.
\end{aligned} \tag{12}$$

In line 4 we used the fact that the probability of sampling label $y$ in $K$ trials without replacement from a pool of $c$ labels is $\frac{K}{c}$.

In line 3 we used the fact that the term $\frac{1}{K}\sum_j \left[\left(p(y|\boldsymbol{x},\mathbf{w}_j) - \frac{1}{K}\sum_i \left(p(y|\boldsymbol{x},\mathbf{w}_i)\right)\right)^2\right]$ only has two possible values.

Let the true label $y$ be in the $K$ sampled labels, specifically let us assume that it is the $a$th sampled label. We can write

$$
\begin{aligned}
&\frac{1}{K}\sum_j \left[\left(p(y|\boldsymbol{x},\mathbf{w}_j) - \frac{1}{K}\sum_i \left(p(y|\boldsymbol{x},\mathbf{w}_i)\right)\right)^2\right] \\
&= \frac{1}{K}\sum_j \left[\left(p(y|\boldsymbol{x},\mathbf{w}_j) - \frac{1}{K}\left(p(y|\boldsymbol{x},\mathbf{w}_a) + \sum_{i\neq a} p(y|\boldsymbol{x},\mathbf{w}_i)\right)\right)^2\right] \\
&= \frac{1}{K}\sum_j \left[\left(p(y|\boldsymbol{x},\mathbf{w}_j) - \frac{1}{K}\left(1+0\right)\right)^2\right] \\
&= \frac{1}{K}\left(\left[\left(p(y|\boldsymbol{x},\mathbf{w}_a) - \frac{1}{K}\right)^2\right] + \sum_{j\neq a}\left[\left(p(y|\boldsymbol{x},\mathbf{w}_j) - \frac{1}{K}\right)^2\right]\right) \\
&= \frac{1}{K}\left(\left[\left(1 - \frac{1}{K}\right)^2\right] + (K-1)\cdot\left[\left(0 - \frac{1}{K}\right)^2\right]\right) \\
&= \frac{K-1}{K^2}.
\end{aligned}
\tag{13}
$$

Now let the true label $y$ *not* be in the $K$ sampled labels. We get

$$
\begin{aligned}
&\frac{1}{K}\sum_j \left[\left(p(y|\boldsymbol{x},\mathbf{w}_j) - \frac{1}{K}\sum_i \left(p(y|\boldsymbol{x},\mathbf{w}_i)\right)\right)^2\right] \\
&= \frac{1}{K}\sum_j \left[\left(p(y|\boldsymbol{x},\mathbf{w}_j) - \frac{1}{K}\cdot 0\right)^2\right] \\
&= \frac{1}{K}\sum_j \left[(0-0)^2\right] \\
&= 0,
\end{aligned}
\tag{14}
$$

where we use the fact that $p(y|\boldsymbol{x},\mathbf{w}_i) = 0,\ \forall i$ if ensemble member $i$ does not fit the true label $y$ but another random label.

□

## E SAMPLING WITH REPLACEMENT

Here we analyze the more complicated case of sampling with replacement. The crucial point is taking into account that the value of the variance term can be cast as the expectation of a function determined only by the number of times we draw the correct class $y$. We then use the fact that being successful $r$ times in $K$ independent trials with a probability $p = \frac{1}{c}$ of success corresponds to a Binomial distribution with parameters $K$ and $p = \frac{1}{c}$.

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

$$\mathbf{E}_{\hat{\rho}\sim\mathcal{A}}\left[\hat{\mathbf{V}}(\hat{\rho})\right] = \frac{1}{2}\left[\sum_r h(r)\binom{K}{r}\left(\frac{1}{c}\right)^r\left(1-\frac{1}{c}\right)^{K-r}\right] \tag{15}$$

*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.*

*Proof.* To analyze this case we need to first assume that given a datasample $(\boldsymbol{x}, y)$ the value of $\frac{1}{K}\sum_j\left[\left(p(y|\boldsymbol{x},\mathbf{w}_j) - \frac{1}{K}\sum_i\left(p(y|\boldsymbol{x},\mathbf{w}_i)\right)\right)^2\right]$ only depends on $r$ the number of times we sample the true label $y$ in $K$ trials with replacement from a pool of $c$ possible labels. Let's then assume that the values are given from a function $h(r)$, it is obvious that what we are evaluating is the expectation of the function $h(r)$ under the Binomial distribution. We get

$$\begin{aligned}
&\mathbf{E}_{\hat{\rho}\sim\mathcal{A}}\left[\hat{\mathbf{V}}(\hat{\rho})\right] = \mathbf{E}_{\hat{\rho}\sim\mathcal{A}}\left[\frac{1}{2m}\sum_U\left[\frac{1}{K}\sum_j\left[\left(p(y|\boldsymbol{x},\mathbf{w}_j) - \frac{1}{K}\sum_i\left(p(y|\boldsymbol{x},\mathbf{w}_i)\right)\right)^2\right]\right]\right] \\
&= \frac{1}{2m}\sum_U\left[\sum_r h(r)\binom{K}{r}\left(\frac{1}{c}\right)^r\left(1-\frac{1}{c}\right)^{K-r}\right] \\
&= \frac{1}{2}\left[\sum_r h(r)\binom{K}{r}\left(\frac{1}{c}\right)^r\left(1-\frac{1}{c}\right)^{K-r}\right]
\end{aligned} \tag{16}$$

where in line 3 we used the fact that the internal expectation is the same for all values in $U$.

To derive the form of $h(r)$ we first assume that given a sample $(\boldsymbol{x}, y)$ only $r$ out of $K$ trials with replacement sample the true label $y$. Denote the set of $r$ ensemble members that fit the true label $y$ as $S$. We then get

$$\begin{aligned}
&\frac{1}{K}\sum_j\left[\left(p(y|\boldsymbol{x},\mathbf{w}_j) - \frac{1}{K}\sum_i\left(p(y|\boldsymbol{x},\mathbf{w}_i)\right)\right)^2\right] \\
&= \frac{1}{K}\sum_j\left[\left(p(y|\boldsymbol{x},\mathbf{w}_j) - \frac{1}{K}\left(\sum_{i\in S}p(y|\boldsymbol{x},\mathbf{w}_i) + \sum_{i\notin S}p(y|\boldsymbol{x},\mathbf{w}_i)\right)\right)^2\right] \\
&= \frac{1}{K}\sum_j\left[\left(p(y|\boldsymbol{x},\mathbf{w}_j) - \frac{1}{K}\left(r\cdot 1 + 0\right)\right)^2\right] \\
&= \frac{1}{K}\left(\sum_{j\in S}\left[\left(p(y|\boldsymbol{x},\mathbf{w}_j) - \frac{r}{K}\right)^2\right] + \sum_{j\notin S}\left[\left(p(y|\boldsymbol{x},\mathbf{w}_j) - \frac{r}{K}\right)^2\right]\right) \\
&= \frac{1}{K}\left[r\cdot\left(1-\frac{r}{K}\right)^2 + (K-r)\cdot\left(0-\frac{r}{K}\right)^2\right] \\
&= \frac{1}{K}\left[r\cdot\left(1-\frac{r}{K}\right)^2 + (K-r)\cdot\left(\frac{r}{K}\right)^2\right] \\
&= h(r)
\end{aligned} \tag{17}$$

□

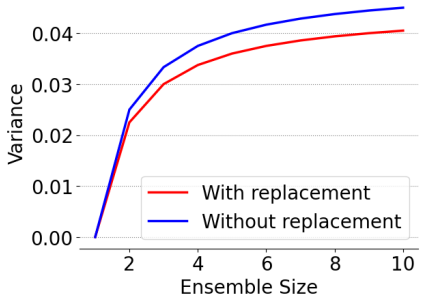



Figure 2: We consider $c = 10$ and $K \in [1, \ldots, 10]$ and plot the variance term $\mathbf{E}_{\hat{\rho}\sim\mathcal{A}}\left[\hat{\mathbf{V}}(\hat{\rho})\right]$ with and without replacement. We see that sampling with replacement results in higher variance for the same number of ensemble members and thus higher diversity for the corresponding ensemble.

Table 1: **With replacement vs without replacement.** We analyze the case of ID performance, 1000 training samples, 10 ensemble members. Sampling the random labels in $\nu$-ensembles *without replacement*, on average results in improvements in calibration metrics such the ECE, the Brier Reliability and the Negative Log-Likelihood. The accuracy and the TACE remain relatively unchanged.

| Dataset / Aug | Method | Acc ↑ | ECE ↓ | TACE ↓ | Brier Rel. ↓ | NLL ↓ | MI ↓ |
|---|---|---|---|---|---|---|---|
| CIFAR-10 / LeNet | **w/ replacement** | 0.510 | 0.137 | 0.030 | 0.120 | 1.680 | **1.236** |
| | **w/o replacement** | **0.514** | **0.131** | **0.028** | **0.117** | **1.650** | 1.245 |
| CIFAR-10 / MLP | **w/ replacement** | **0.401** | **0.083** | **0.022** | **0.087** | **1.753** | 1.650 |
| | **w/o replacement** | **0.401** | 0.098 | 0.023 | 0.092 | 1.767 | **1.559** |
| CIFAR-10 / ResNet22 | **w/ replacement** | 0.520 | 0.016 | 0.020 | 0.090 | 1.471 | 0.699 |
| | **w/o replacement** | **0.525** | **0.013** | **0.018** | **0.087** | **1.460** | **0.691** |
| CIFAR-100 / LeNet | **w/ replacement** | 0.145 | 0.220 | **0.006** | 0.151 | 5.343 | 1.988 |
| | **w/o replacement** | **0.147** | **0.155** | **0.006** | **0.113** | **4.846** | **1.654** |
| CIFAR-100 / MLP | **w/ replacement** | **0.103** | 0.116 | 0.005 | 0.078 | 4.447 | 3.047 |
| | **w/o replacement** | **0.103** | **0.040** | **0.004** | **0.049** | **4.171** | **2.807** |
| CIFAR-100 / ResNet22 | **w/ replacement** | **0.134** | **0.093** | **0.006** | **0.074** | **4.266** | **1.086** |
| | **w/o replacement** | **0.134** | 0.135 | **0.006** | 0.099 | 4.892 | 1.476 |
| mean diff. | w/o - w/ | **-0.001** | **-0.015** | **-0.0006** | **-0.010** | **-0.033** | **-0.031** |

## F SAMPLING WITH REPLACEMENT VS SAMPLING WITHOUT REPLACEMENT

We first plot the theoretical variance when sampling with replacement compared to when sampling without replacement. We see that sampling with replacement results in higher variance for the same number of ensemble members and thus higher diversity for the corresponding ensemble.

We perform the CIFAR-10 and CIFAR-100 experiments with an ensemble size of 10, using sampling with and without replacement and compare the results in Table 1. In the last row we compute the average difference between the metrics when sampling without replacement compared to sampling with replacement. We see that on average sampling without replacement results in improvements across different calibration metrics such as the ECE, Brier Reliability and Negative Log-Likelihood. The accuracy and the TACE remain relatively unchanged. At the same time the diversity of the ensemble also improves. These results validate our theoretical analysis, and further motivate improving the ensemble diversity using labels sampled without replacement.

Table 2: **ID performance, 1000 training samples, 10 ensemble members.** $\nu$-ensembles retain approximately the same accuracy as standard ensembles. At the same time, they achieve significantly better calibration in all calibration metrics. These results are consistent with the experiments for the CIFAR-10 and the CIFAR-100. The only outlier is the ResNet22 architecture for the STL10 dataset, where $\nu$-ensembles underfit the data.

| Dataset / Aug | Method | Acc ↑ | ECE ↓ | TACE ↓ | Brier Rel. ↓ | NLL ↓ | MI ↓ |
|---|---|---|---|---|---|---|---|
| SVHN / LeNet | Standard | **0.618** | 0.138 | 0.028 | 0.114 | 1.679 | **1.561** |
| | **$\nu$-ensembles** | 0.605 | **0.083** | **0.023** | **0.102** | **1.371** | 1.792 |
| SVHN / MLP | Standard | **0.474** | 0.252 | 0.047 | 0.170 | 2.748 | 1.733 |
| | **$\nu$-ensembles** | 0.471 | **0.157** | **0.037** | **0.128** | **2.008** | **1.653** |
| SVHN / ResNet22 | Standard | **0.707** | **0.070** | **0.019** | 0.098 | 1.012 | 0.921 |
| | **$\nu$-ensembles** | 0.700 | **0.070** | **0.019** | **0.096** | **0.988** | **0.906** |
| STL10 / LeNet | Standard | 0.309 | 0.045 | 0.018 | 0.051 | 1.91 | **1.821** |
| | **$\nu$-ensembles** | **0.310** | **0.020** | **0.017** | **0.043** | **1.896** | 1.854 |
| STL10 / MLP | Standard | **0.302** | 0.021 | 0.016 | 0.037 | 1.905 | 1.714 |
| | **$\nu$-ensembles** | **0.302** | **0.013** | **0.015** | **0.033** | **1.897** | **1.660** |
| STL10 / ResNet22 | Standard | **0.302** | **0.037** | **0.018** | **0.045** | **1.898** | **0.865** |
| | **$\nu$-ensembles** | 0.278 | 0.217 | 0.050 | 0.134 | 2.423 | 0.642 |

## G ADDITIONAL DATASETS

Here we explore additional datasets, the SVHN dataset (Buitinck et al., 2013) and the STL10 dataset (Coates et al., 2011). We use 1000 training samples, 3000 validation samples, 1000 unlabeled samples and the original test sets for both datasets. We plot the results in table 2. We see that on average the results much those for the CIFAR-10 and CIFAR-100 case. $\nu$-ensembles achieve improvements in calibration while typically not hurting accuracy.

## H EXPERIMENTS WITH MULTIPLE SEEDS

Here we conduct experiments with multiple seeds so as to ensure that our results are robust. Specifically, we repeat the experiments for Standard and $\nu$-ensembles on CIFAR-10 and CIFAR-100 with 3 seeds for each ensemble member and plot the results in 3. We see that the results are approximately the same as single seed experiments.

## I EXPERIMENTAL SETUP

We ran all experiments using A100, and V100 NVIDIA GPUs on our cluster. In total, the experiments consumed approximately 10000 hours of GPU time. The implementations were done in JAX Bradbury et al. (2018). While data loading was done in Tensorflow Abadi et al. (2015). For $\nu$-ensembles, for the LeNet architecture we investigated epochs in the range $[100, 120, 140, 160, 180, 200, 220, 240, 260]$, for the MLP $[100, 120, 140, 160, 180, 200, 220, 240, 260]$, for the ResNet $[200, 220, 250, 270, 300, 320, 350, 370, 400]$. For the regularization strength, we searched in the range $[1, 0.1, 0.05, 0.01, 0]$ and for the optimizer learning rate in $[0.0001, 0.001]$. We investigated the same epoch and learning rate ranges for Standard ensembles. Agree to Disagree ensembles contain a single hyperparameter $\alpha$. We tested values in the range $[1, 0.1, 0.01, 0.001, 0.0001]$.

Table 3: **ID performance, 1000 training samples, 10 ensemble members, 3 seeds.** $\nu$-ensembles retain the same performance as the single seed experiments.

| 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 |
| | **$\nu$-ensembles** | 0.506 | 0.133 | 0.028 | 0.118 | 1.664 | 1.201 |
| CIFAR-10 / MLP | Standard | 0.399 | 0.205 | 0.043 | 0.144 | 2.078 | 1.622 |
| | **$\nu$-ensembles** | 0.399 | 0.086 | 0.023 | 0.087 | 1.782 | 1.525 |
| CIFAR-10 / ResNet22 | Standard | 0.527 | 0.087 | 0.024 | 0.106 | 1.690 | 0.939 |
| | **$\nu$-ensembles** | 0.527 | 0.014 | 0.017 | 0.082 | 1.436 | 0.675 |
| CIFAR-100 / LeNet | Standard | 0.149 | 0.300 | 0.007 | 0.212 | 8.817 | 2.276 |
| | **$\nu$-ensembles** | 0.147 | 0.186 | 0.006 | 0.131 | 5.115 | 1.826 |
| CIFAR-100 / MLP | Standard | 0.101 | 0.183 | 0.007 | 0.114 | 5.173 | 3.142 |
| | **$\nu$-ensembles** | **0.103** | 0.156 | 0.006 | 0.106 | 4.906 | 3.014 |
| CIFAR-100 / ResNet22 | Standard | 0.137 | 0.196 | 0.007 | 0.141 | 7.810 | 1.688 |
| | **$\nu$-ensembles** | 0.135 | 0.135 | 0.006 | 0.099 | 4.922 | 1.475 |

## REFERENCES

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. URL https://www.tensorflow.org/. Software available from tensorflow.org.

Pierre Alquier, James Ridgway, and Nicolas Chopin. On the properties of variational approximations of Gibbs posteriors. *The Journal of Machine Learning Research*, 17(1):8374–8414, 2016.

James Bradbury, Roy Frostig, Peter Hawkins, Matthew James Johnson, Chris Leary, Dougal Maclaurin, George Necula, Adam Paszke, Jake VanderPlas, Skye Wanderman-Milne, and Qiao Zhang. JAX: composable transformations of Python+NumPy programs, 2018. URL http://github.com/google/jax.

Lars Buitinck, Gilles Louppe, Mathieu Blondel, Fabian Pedregosa, Andreas Mueller, Olivier Grisel, Vlad Niculae, Peter Prettenhofer, Alexandre Gramfort, Jaques Grobler, Robert Layton, Jake VanderPlas, Arnaud Joly, Brian Holt, and Gaël Varoquaux. API design for machine learning software: experiences from the scikit-learn project. In *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*, pp. 108–122, 2013.

Adam Coates, Andrew Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pp. 215–223. JMLR Workshop and Conference Proceedings, 2011.

Gintare Karolina Dziugaite and Daniel M Roy. Computing nonvacuous generalization bounds for deep (stochastic) neural networks with many more parameters than training data. *Uncertainty in Artificial Intelligence*, 2017.

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

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.