Why Are Bootstrapped Deep Ensembles Not Better?

Jeremy Nixon
Google Research, Brain Team
jeremynixon@google.com

Dustin Tran
Google Research, Brain Team
trandustin@google.com

Balaji Lakshminarayanan
Google Research, Brain Team
balajiln@google.com

Abstract

Ensemble methods have consistently reached state of the art across predictive, uncertainty, and out-of-distribution robustness benchmarks. One of the most popular ways to construct an ensemble is to independently train each model on a resampled (bootstrapped) version of the dataset. Bootstrapping is popular in the literature on decision trees and frequentist statistics, with strong theoretical guarantees, but it is not used often in practice for deep neural networks. We investigate a common hypothesis for bootstrap's weak performance—percentage of unique points in the subsampled dataset—and find that even when adjusting for it, bootstrap ensembles of deep neural networks yield no benefit over simpler baselines. This brings to question the role of data randomization as a source of uncertainty in deep learning.

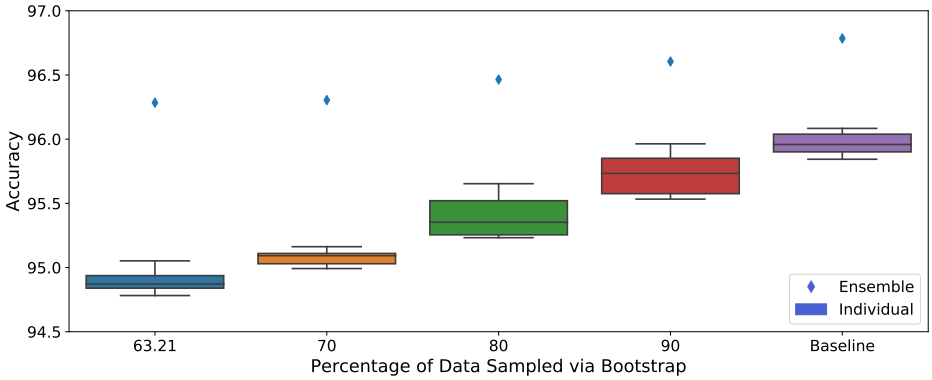

Figure 1: Bootstrap ensemble of Wide ResNet 28-10 for CIFAR-10. Box plots display the 10 individual models in the ensemble. With bootstrap, each model trains on only 63.21% unique elements of the original dataset, leading to poor performance. Even if we resample the dataset until a certain percentage, we can only recover the baseline and not improve it. This suggests that bootstrap's data randomization may not yield any benefit for deep learning.

1 Introduction

Let $\boldsymbol{x}_n \in \mathbb{R}^D$ denote D-dimensional features and $y_n \in [1, \ldots, K]$ denote the corresponding class label. Assume we have a parametric model $p(y|\boldsymbol{x}, \boldsymbol{\theta})$ for the conditional distribution, and a prior $p(\boldsymbol{\theta})$ over parameters. The Bayesian posterior over parameters is given by

$$p(\boldsymbol{\theta}|\{\boldsymbol{x}_n, y_n\}_{n=1}^N) \propto p(\boldsymbol{\theta}) \prod_{n=1}^N p(y_n|\boldsymbol{x}_n, \boldsymbol{\theta}). \tag{1}$$

1st I Can't Believe It's Not Better Workshop (ICBINB@NeurIPS 2020), Vancouver, Canada.

Computing the posterior over $\boldsymbol{\theta}$ is computationally challenging when $p(y_n|\boldsymbol{x}_n, \boldsymbol{\theta})$ is parametrized using a deep neural network. While computing the posterior is challenging, it is usually easy to compute the MAP solution, which corresponds to the mode of the posterior. The MAP solution can be written as the minimizer of the following loss (negative log likelihood + negative log prior):

$$L(\boldsymbol{\theta}, \{\boldsymbol{x}_n, y_n\}_{n=1}^N) = -\log p(\boldsymbol{\theta}) - \sum_{n=1}^N \log p(y_n|\boldsymbol{x}_n, \boldsymbol{\theta}). \tag{2}$$

$$\hat{\boldsymbol{\theta}}_{\mathsf{MAP}} = \arg\min_{\boldsymbol{\theta}} L(\boldsymbol{\theta}, \{\boldsymbol{x}_n, y_n\}_{n=1}^N). \tag{3}$$

The MAP solution is computationally efficient, but only gives a point estimate and not a distribution over parameters. There are two popular ways to extend MAP to get multiple solutions:

- *Multiple random seeds*: Initialize at M different values for each ensemble member, possibly with different minibatch ordering, and repeat the minimization in equation 2. This can produce M different solutions if the loss is non-convex.
- *Bootstrap aggregation a.k.a. Bagging* [Breiman, 1996]: Train members on M different bootstrap samples of the original dataset.

Concretely, bagging first samples a "bootstrap dataset" of N points with replacement from the original training set of N unique points, and then trains a neural network on this resampled dataset. Conceptually, it is equivalent to drawing a set of instance weights $\boldsymbol{w} = [w_1, \ldots, w_N]$ from a Multinomial distribution N times with mean parameter $[1/N, ..., 1/N]$. Note that $\sum_n w_n = N$.

Training a neural network on the bootstrap dataset is equivalent to minimizing the weighted loss function:

$$L(\boldsymbol{\theta}, \{\boldsymbol{x}_n, y_n\}_{n=1}^N, \boldsymbol{w}) = -\log p(\boldsymbol{\theta}) - \sum_{n=1}^N w_n \log p(y_n|\boldsymbol{x}_n, \boldsymbol{\theta}). \tag{4}$$

$$\hat{\boldsymbol{\theta}}_{\mathsf{WMAP}} = \arg\min_{\boldsymbol{\theta}} L(\boldsymbol{\theta}, \{\boldsymbol{x}_n, y_n\}_{n=1}^N, \boldsymbol{w}). \tag{5}$$

where $\hat{\boldsymbol{\theta}}_{\mathsf{WMAP}}$ denotes the "weighted MAP" solution. Note that setting $w_n = 1 \forall n$ in equation 4 recovers the usual unweighted case in equation 2.

Bootstrap is theoretically well-motivated [Efron, 1992] and popular in the literature on both ensembles of decision trees (cf. random forests [Breiman, 2001]) and in uncertainty estimation for calibrated confidence intervals, but it has been found to hurt performance for ensembles of deep neural networks. Lakshminarayanan et al. [2017], Lee et al. [2015] report that using the entire dataset for all the networks in the ensemble, which corresponds to a fixed value of $\boldsymbol{w} = [1, \ldots, 1]$, works better than training the individual networks on bootstrap versions of the original dataset. Lakshminarayanan et al. [2017] hypothesize that the poor empirical performance could be due to the fact that the bootstrap sample only contains 63.2% of unique points.[1] Earlier work on random forests such as Breiman's random forests [Breiman, 2001] also used bootstrap, while later work such as extremely randomized trees (ERT) [Geurts et al., 2006] found that bootstrap was not necessary when using other sources of randomization.

We investigate this hypothesis by implementing an extension of bootstrap that increases the percentage of unique points. We then evaluate the ensembles' accuracy and calibration on Fashion MNIST, CIFAR-10, and CIFAR-100. We find that increasing the percentage does improve bootstrap's performance. Ultimately, it is not the sole factor: even when close to using 100% unique points, bootstrap does not provide any improvements over well-tuned baselines.

2 Experiments

We evaluate a Wide-Resnet [Zagoruyko and Komodakis, 2016] ensemble on CIFAR-100 and CIFAR-10 [Krizhevsky, 2009], as well as a small CNN ensemble on Fashion-MNIST [Xiao et al.,

[1] From Lakshminarayanan et al. [2017, §2.4]: *"The bootstrap draws N times uniformly with replacement from a dataset with N items. The probability an item is picked at least once is $1 - (1 - 1/N)^N$, which for large N becomes $1 - e^{-1} \approx 0.632$. So, the expected number of unique data points in a bootstrap sample is $0.632N$."*

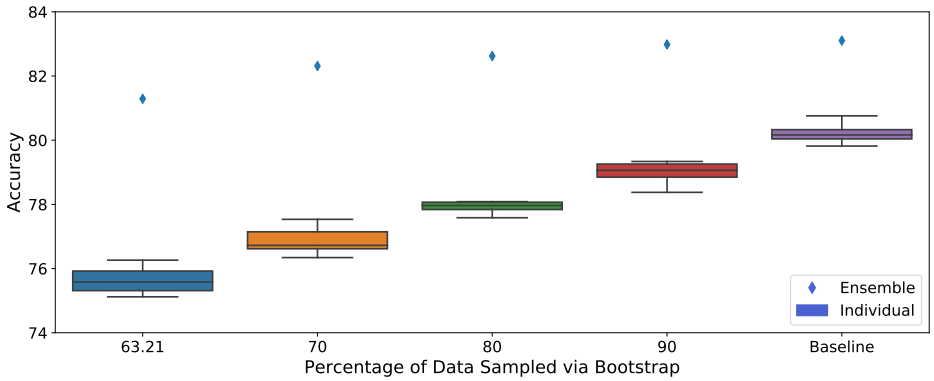

Figure 2: Bootstrap ensemble of Wide ResNet 28-10 for CIFAR-100. The results are consistent with Figure 1, which illustrate bootstrap for CIFAR-10.

2017]. We evaluate accuracy and calibration of individual networks as well as ensembles. All ensembles trained have 10 members, unless indicated otherwise.

2.1 Evaluation

We evaluate calibration using the expected calibration error (ECE) [Naeini et al., 2015].

Expected Calibration Error (ECE). To approximate the calibration error in expectation, ECE Naeini et al. [2015] discretizes the probability interval into a fixed number of bins, and assigns each predicted probability to the bin that encompasses it. The calibration error is the difference between the fraction of predictions in the bin that are correct (accuracy) and the mean of the probabilities in the bin (confidence). Intuitively, the accuracy estimates $\mathbb{P}(Y = y \mid \hat{p} = p)$, and the average confidence is a setting of p. ECE computes a weighted average of this error across bins:

$$\mathrm{ECE} = \sum_{b=1}^{B} \frac{n_b}{N} \left|\mathrm{acc}(b) - \mathrm{conf}(b)\right|,$$

where n_b is the number of predictions in bin b, N is the total number of data points, and $\mathrm{acc}(b)$ and $\mathrm{conf}(b)$ are the accuracy and confidence of bin b, respectively.

2.2 Supersampling the dataset does not lead to improved performance

We evaluate a Wide Resnet on data from bootstrap draws from a dataset as well as on the standard dataset. As mentioned earlier, the bootstrap sampled data contains 63.2% unique points. We evaluate reweighting by supersampling the training dataset for each ensemble member. The bootstrap is modified to sample data with replacement until a given percentage of the training dataset has been sampled. We sample up to 70%, 80%, 90%, and 99% of the training data. In this case diversity is in the form of a reweighting of the dataset which comes out of the randomness in the datapoint selection. Finally, we also report results for deep ensembles [Lakshminarayanan et al., 2017] which correspond to the case where each ensemble member is trained on the entire 100% of the dataset, and the source of diversity is random initialization and randomness in SGD [Fort et al., 2019].

In Figure 1 we show the distribution of ensemble member performance alongside ensemble performance in a box-and-whisker plot on CIFAR-10 and CIFAR-100. The median ensemble member's accuracy is marked by the orange line, with the minimum and maximum ensemble members marked by a dot or an end line depending on whether they are within 1.5 * the Inter Quartile Range (25th-75th percentile range) of the data. Models trained on bootstrap dataset (left most point corresponding to 0.632 marker) achieve a much smaller accuracy on their validation set than models trained on the full dataset, which is to be expected. While the ensemble of the bootstrap sample members performs better than the single model, the gains from ensembling do not overcome the drop in accuracy due to dataset size, underperforming an ensemble trained on the full dataset. As the number of unique

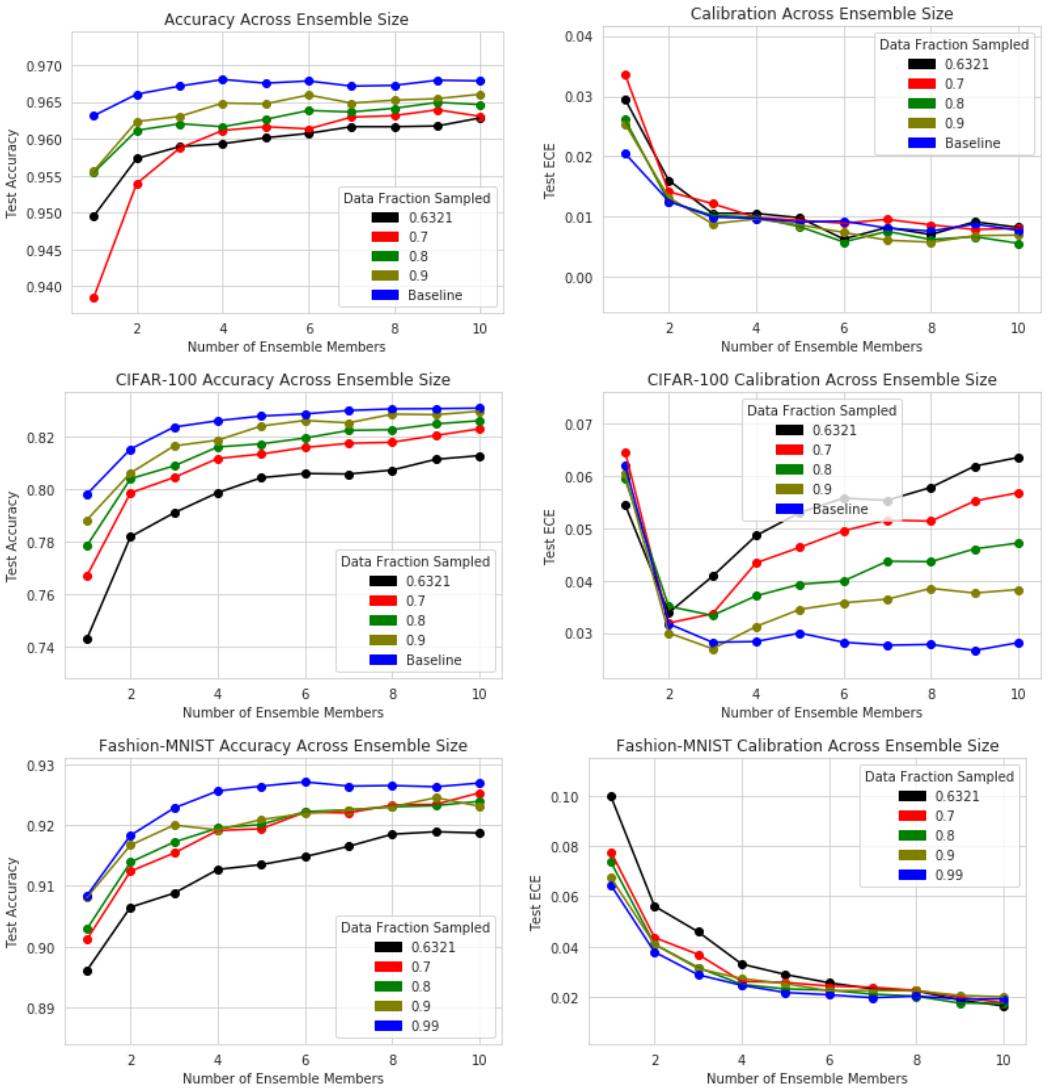

Figure 3: Bootstrap ensembles are dominated by standard deep ensembles on both accuracy and calibration error across the number of ensemble members. The baseline ensemble at 1.0 trains on the standard dataset which is not supersampled.

datapoints in the training dataset increases, both the individual ensemble member performance and overall ensemble performance consistently improve peaking at the use of the full dataset.

In Figure 3 we investigate ensemble size and its impact on calibration error and accuracy for varying fraction of unique datapoints. The trend is that ensembles trained with the bootstrap are dominated in accuracy by ensembles trained without it at every number of ensemble members from 2 up to 10. We also observe that deep ensembles require fewer networks to achieve the same performance. With calibration, the impact of full dataset training is not as dramatic. While the general trend is also for ensembles trained with more unique training datapoints to outperform models trained with a heavier bootstrap, CIFAR 10 trained with > 4 ensemble members and Fashion-MNIST trained with > 7 members show a negligible difference between ECE of ensembles trained on a varying number of unique samples. We also notice an interesting phenomenon with ECE on CIFAR-100 merits further investigation: where deep ensembles are better calibrated with increasing members but bootstrap ensembles are better calibrated for $M = 2$ but their ECE becomes worse for higher M.

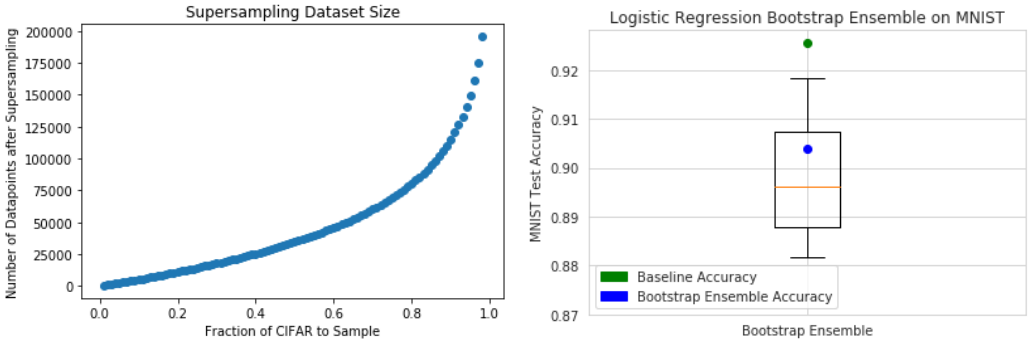

Figure 4: Left: Ratio of fraction of the dataset to the total number of datapoints sampled under supersampling. Right: Logistic regression trained with a bootstrap ensemble does not outperform, as a linear model's variance is already very low. Te Baseline Accuracy uses the full training dataset.

2.3 Ensemble Calibration Evaluation

On CIFAR-100 we observe a number of surprising calibration phenomena, pictured in Figure 5. As the ensemble size increases, the model becomes increasingly overconfident, especially in the standard bootstrap setting where the fraction of unique datapoints is 63.21%.

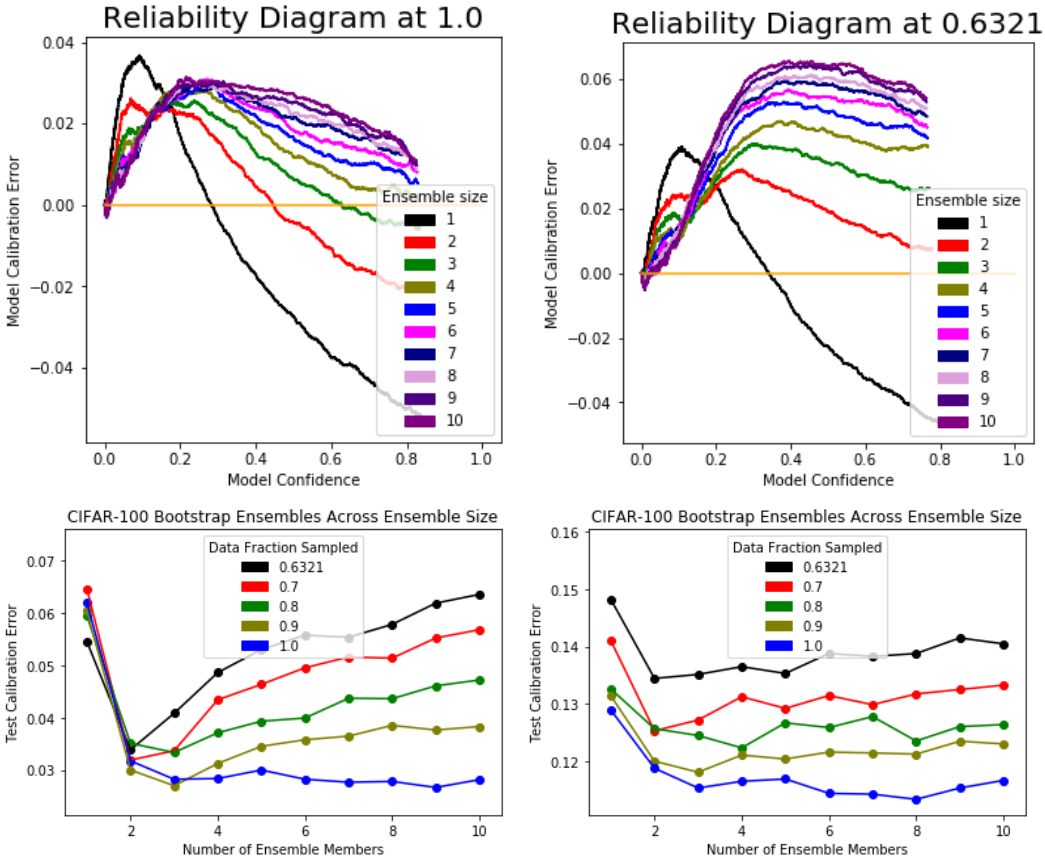

Figure 5: Interesting calibration phenomena on CIFAR-100. Top: Calibration error against model confidence reliability diagram as the ensemble size increases. Bottom: ECE (Left) compared with Class-Conditional ECE (Right). In ECE predictions for different classes are merged and then the calibration error is computed. In Class-Conditional ECE classes have the calibration error computed independently and then an average is taken.

2.4 Proper Scoring Rule Evaluation

The Brier score and cross-entropy loss are robust measures of model performance. We note in Figure 6 that as the number of unique data points sampled and trained on by our ensemble members increases (with supersampling) we see improvements in the Cross Entropy and in the Brier score. Our proper scoring rules also improve as the number of ensemble members improve (in contrast with the calibration error on CIFAR-100 in Figure 1.

include a measure of reliability

Note correlation between the two.

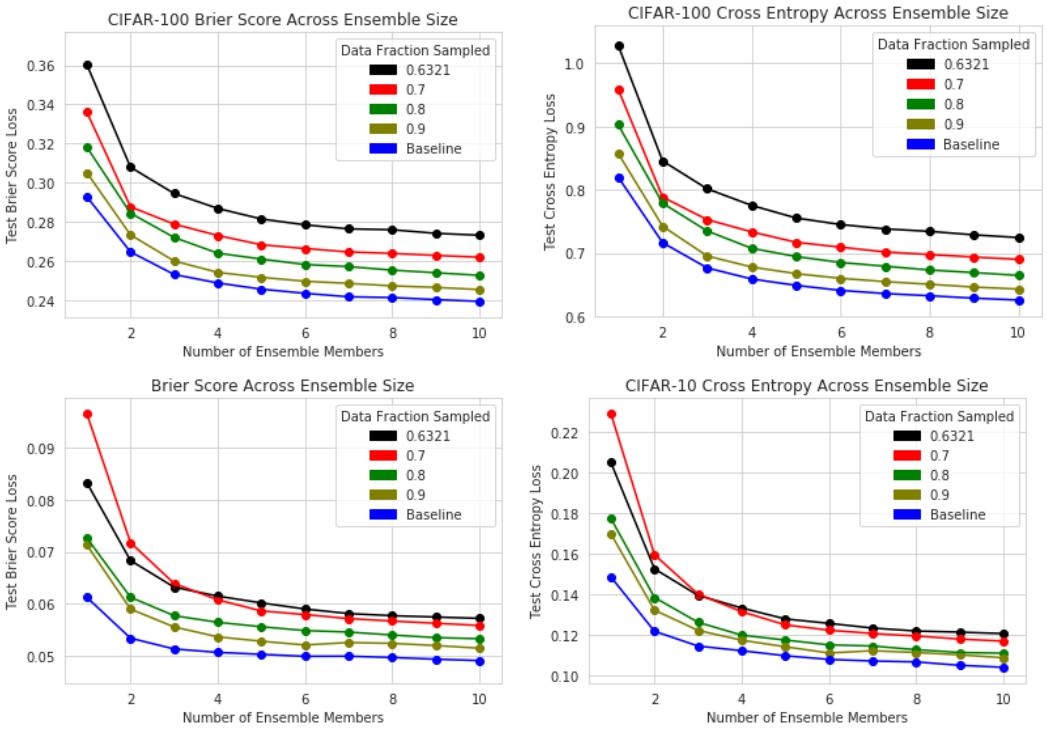

Figure 6: As bootstrap ensemble size increases, proper scoring rule performance consistently improves.

2.5 Supersampling vs. Standard Sampling

2.6 Ablation: Sources of Randomness

One natural question about using the bootstrap as a source of variability for ensembling is to ask whether the diversity vale from the bootstrap is already captured by existing forms of variability. There are three other major sources of variability - random initializaiton, stochastic gradient descent's batch choices, and stochasticity in tensorflow operations used in training. In Figure 7 we show the surprisingly high quality performance of ensemble models trained using the same initialization.

3 Discussion

We investigate the effect of bootstrapping strategies on the accuracy and calibration of ensembles. We find that the number of unique points significantly affects the accuracy of the ensemble as well as individual members. Using the entire dataset typically performs best w.r.t. calibration, and interestingly, bootstrap ensembles with fewer unique data points can still be well-calibrated. For future work, we plan to look at calibration under dataset shift [Ovadia et al., 2019] and detecting out-of-distribution inputs [Hendrycks and Gimpel, 2017, Lakshminarayanan et al., 2017].

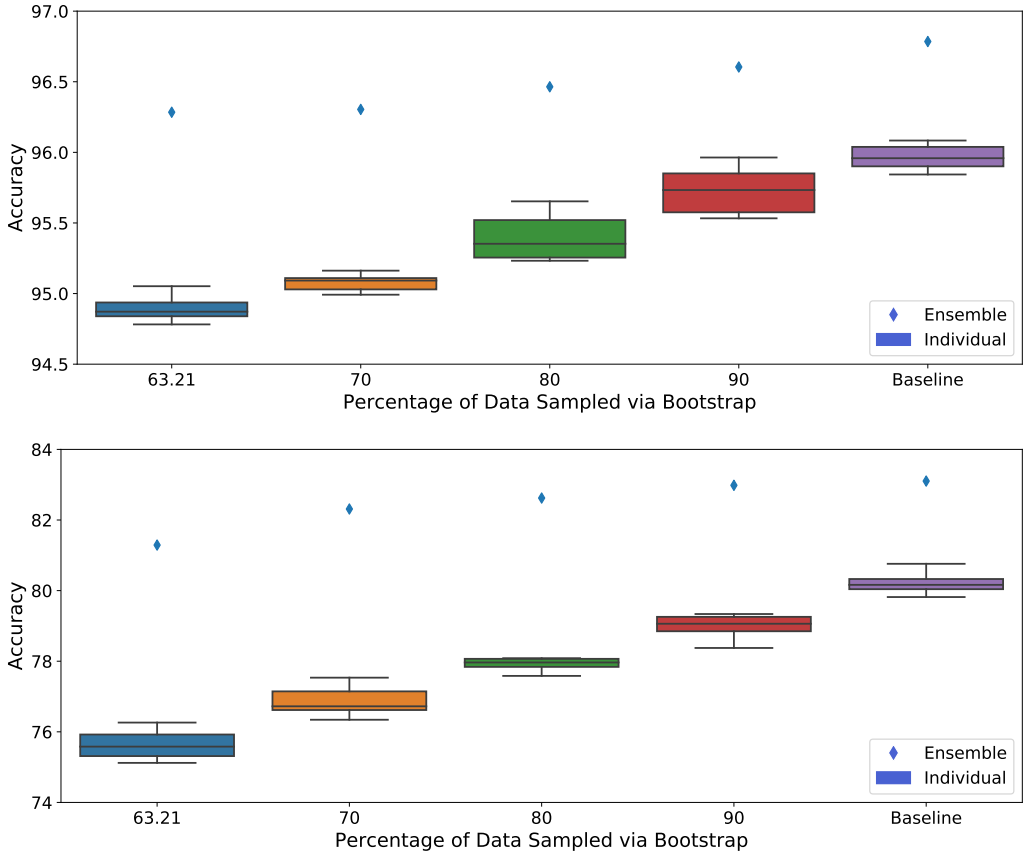

Figure 7: Ablation which controls the **initialization** of the model on CIFAR-10 (Top) and CIFAR-100 (Bottom). All ensemble memebers are trained with the same initialization. Ensemble diversity and performance quality is preserved.

References

L. Breiman. Random forests. *Machine learning*, 2001.

Leo Breiman. Bagging predictors. *Machine learning*, 24(2):123–140, 1996.

Bradley Efron. Bootstrap methods: another look at the jackknife. In *Breakthroughs in statistics*, pages 569–593. Springer, 1992.

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

Pierre Geurts, Damien Ernst, and Louis Wehenkel. Extremely randomized trees. *Machine learning*, 63(1):3–42, 2006.

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

Alex Krizhevsky. Learning multiple layers of features from tiny images. 2009.

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

Stefan Lee, Senthil Purushwalkam, Michael Cogswell, David Crandall, and Dhruv Batra. Why M heads are better than one: Training a diverse ensemble of deep networks. *arXiv preprint arXiv:1511.06314*, 2015.

Mahdi Pakdaman Naeini, Gregory F Cooper, and Milos Hauskrecht. Obtaining well calibrated probabilities using Bayesian binning. In *AAAI*, pages 2901–2907, 2015.

Yaniv Ovadia, Emily Fertig, Jie Ren, Zachary Nado, D Sculley, Sebastian Nowozin, Joshua V Dillon, Balaji Lakshminarayanan, and Jasper Snoek. Can you trust your model's uncertainty? Evaluating predictive uncertainty under dataset shift. In *NeurIPS*, 2019.

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016.