Deep Ensembles Work, But Are They Necessary?

Taiga Abe[*1] **E. Kelly Buchanan**[*1] **Geoff Pleiss**[1] **Richard Zemel**[1]

John P. Cunningham[1]
[1]Columbia University
{ta2507,ekb2154,gmp2162,jpc2181}@columbia.edu
zemel@cs.columbia.edu

Abstract

Ensembling neural networks is an effective way to increase accuracy, and can often match the performance of individual larger models. This observation poses a natural question: given the choice between a deep ensemble and a single neural network with similar accuracy, is one preferable over the other? Recent work suggests that deep ensembles may offer distinct benefits beyond predictive power: namely, uncertainty quantification and robustness to dataset shift. In this work, we demonstrate limitations to these purported benefits, and show that a single (but larger) neural network can replicate these qualities. First, we show that ensemble diversity, by any metric, does not meaningfully contribute to an ensemble's uncertainty quantification on out-of-distribution (OOD) data, but is instead highly correlated with the relative improvement of a single larger model. Second, we show that the OOD performance afforded by ensembles is strongly determined by their in-distribution (InD) performance, and—in this sense—is not indicative of any "effective robustness." While deep ensembles are a practical way to achieve improvements to predictive power, uncertainty quantification, and robustness, our results show that these improvements can be replicated by a (larger) single model.

1 Introduction

In many real-world settings, practitioners deploy ensembles of neural networks that combine the outputs of several individual models [e.g. 71, 44, 77]. Though training and evaluating multiple models is computationally expensive, a wide body of research demonstrates that ensembles achieve better performance (as measured by accuracy, negative log likelihood, or a variety of other metrics) than their constituent single models, provided that these models make diverse errors [16]. This benefit is well-established in the literature: theoretically proven for ensembles formed via boosting or bagging [67, 6], and demonstrated for *deep ensembles* that solely rely on the randomness of SGD coupled with non-convex loss surfaces [46, 18].

Of course, ensembling is not the only way to increase performance; one could also increase the depth or width of a single neural network. In many settings, a single large model performs similarly to an ensemble of (smaller) models with a similar number of parameters [48, 40, 74]. This observation poses a natural question: are there reasons to choose a deep ensemble over a single (larger) neural network with comparable performance?

Recent research suggests that deep ensembles may be preferable to single models in safety-critical applications and settings where data shifts significantly away from the training distribution. First,

[*]Equal contribution.

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

Lakshminarayanan et al. [45] demonstrate that deep ensembles provide *well-calibrated estimates of uncertainty* on classification and regression tasks. Compared with other uncertainty quantification (UQ) methods, ensembles offer better (i.e. less overconfident) uncertainty estimates on out-of-distribution (OOD) or shifted data [62]. Second, recent work indicates that—beyond calibration—ensemble performance (as measured by accuracy, NLL, or other metrics) also tends to be *robust against dataset shift*, again often outperforming other methods in these regimes [27].

Intuitions in recent papers [e.g. 46, 18] attribute these UQ/robustness benefits to the fact that ensembles produce multiple diverse predictions, rather than a single point prediction. If diversity does in fact explain UQ/robustness improvements, this would suggest that deep ensembles indeed offer benefits that cannot be obtained by (standard) single neural networks. In this paper, we rigorously test hypotheses that formalize this intuition. Surprisingly, after controlling for factors related to the performance of an ensemble's component models, we find no evidence that having a diverse set of predictions is responsible for these purported benefits. Put differently, we find that these UQ/robustness benefits are not unique to deep ensembles, *as they can be replicated through the use of (larger) single models*. We confirm these results for a wide variety of model architectures, as well as for *heterogeneous deep ensembles* that combine multiple different neural network architectures and *implicit deep ensembles* like MC Dropout [21], BatchEnsemble [75], and MIMO [30] (Appx. H.4).

Hypothesis: ensemble diversity is responsible for improved UQ. Two components contribute to ensemble uncertainty estimates: the uncertainties expressed by individual ensemble members, and diversity among ensemble member predictions. Recent work suggests that the diversity component is primarily responsible for better calibrated OOD uncertainty estimates, as ensemble members should agree less (i.e. offer more diverse predictions) as data shift away from the training distribution [45, 18, 27]. In contrast, we find that—after conditioning on the uncertainty of individual ensemble members—*the level of ensemble disagreement does not statistically differ between InD and OOD data* (Fig. 1), and thus ensemble diversity is not directly responsible for larger OOD uncertainty estimates. Furthermore, ensemble diversity—on a per-datapoint basis—is correlated with the expected improvement we obtain by increasing model capacity (Fig. 2), implying that ensemble diversity does not capture a quantity inaccessible to a single (larger) model.

Hypothesis: ensemble diversity is responsible for improved robustness. Independent work demonstrates a deterministic relationship between a (single) neural network's 0-1 accuracy on InD and OOD datasets [72, 52], whereby the OOD performance of a model can be predicted from its InD performance. It is therefore natural to ask whether having multiple diverse predictions contributes to additional OOD robustness (as suggested by [18, 27]), beyond what is expected given performance improvements on InD data. Our results demonstrate that deep ensembles are not "effectively robust" relative to single models—i.e. their OOD performance (as measured by accuracy, NLL, Brier score, and calibration error) follows the same deterministic relationship to InD performance as single models (Fig. 4). Therefore, ensemble diversity does not yield additional robustness over what standard single networks achieve.

Implications. Overall, this paper does not disagree with prior claims about the benefits of deep ensembles relative to an ensemble's component models. Indeed, in our experiments we confirm that ensembling is a convenient mechanism to improve predictive performance, UQ, and robustness relative to this baseline. At the same time, our results also indicate that—after controlling for individual model uncertainty and InD performance— ensembles do not obtain UQ/robustness benefits beyond what can already be obtained from the properties of an appropriately chosen single model.

2 Related work

Ensembling is an established technique to improve generalization [e.g. 67, 63, 17, 60], where the predictions of multiple models are aggregated to reach a consensus. It is well established that diversity amongst ensemble members is necessary to improve performance [16]. This diversity can be achieved through many means. Randomization approaches introduce diversity by training each model on a random subset of data [6] or a random subset of features [7]. Alternatively, boosting approaches [19, 20] achieve diversity by manipulating the weighting of training data. Other methods include using a diverse set of model classes [e.g. 10] or joint training objectives [e.g. 55].

Ensembles of neural networks. Historically, neural network ensembles have relied on a variety of mechanisms to introduce diversity [e.g. 28, 63, 54, 79]. Recently, diversity is often obtained by

training multiple copies of the same neural network architecture with different intializations and minibatch orderings, as the inherent randomness of SGD has been shown to introduce a sufficient amount of diversity in these (non-convex) models [46, 24, 18]. Importantly, this approach can exploit parallel computation [45], because none of the ensemble members depend on one another.

Deep ensembles for predictive uncertainty. It has been suggested that ensembles of neural networks not only improve accuracy but also estimates of predictive uncertainty [45]. Some research aims to connect ensembles and Bayesian neural networks, suggesting that these improved uncertainty estimates are the result of performing approximate Bayesian model averaging [21, 76]. Although prior work has described shortcomings in the uncertainty estimates derived from deep ensembles [e.g. 47, 11, 32, 61], they remain a gold standard in high risk and safety critical settings [e.g. 62, 27, 73].

Deep ensembles and robustness. Robustness is the ability to maintain good accuracy and calibration under conditions of distributional shift. Deep ensembles outperform other approaches in maintaining both accuracy and calibration on OOD data [62, 27], although their limitations have also been demonstrated [43, 64]. This robustness is attributed to the diversity between ensemble members [18].

Other related work. Recent work investigates whether it is possible to achieve the benefits of an ensemble with reduced computation during training and/or test time [36, 49, 75, 30]. Additionally many works have proposed numerous diversity metrics for ensembles similar to those we examine here [e.g. 42, 51, 3].

3 Setup

Consider multiclass classification: inputs $\boldsymbol{x} \in \mathbb{R}^D$ with targets $y \in [1, \ldots, C]$, where D is the number of features and C is the number of classes. We assume that we have access to M distinct neural networks $\boldsymbol{f}_1, \ldots, \boldsymbol{f}_M$, where each model $\boldsymbol{f}_i : \mathbb{R}^D \to \Delta^C$ maps an input to the C-class probability simplex. We will primarily focus on the common case of **homogeneous ensembles**, where $\boldsymbol{f}_1, \ldots, \boldsymbol{f}_M$ represent the same neural network architecture and training procedure, relying on the inherent randomness of initialization and SGD to produce diverse models (see Sec. 2 for a broad discussion). However, in Sec. 5.3 we will also consider **heterogeneous ensembles** where $\boldsymbol{f}_1, \ldots, \boldsymbol{f}_M$ represent different architectures or training procedures, and **implicit ensembles**, where $\boldsymbol{f}_1, \ldots, \boldsymbol{f}_M$ are approximated by changes to a single model [21, 30, 75]. Throughout the paper, we will also represent these member networks as a discrete distribution of models: $p(\boldsymbol{f}) = \text{Unif.}[\boldsymbol{f}_1, \ldots, \boldsymbol{f}_M]$. The ensemble prediction $\bar{\boldsymbol{f}}(\boldsymbol{x})$ is given by the arithmetic mean of the ensemble member *probabilities*:[1]

$$\bar{\boldsymbol{f}}(\boldsymbol{x}) = \mathbb{E}_{p(\boldsymbol{f})}[\boldsymbol{f}(\boldsymbol{x})] = \tfrac{1}{M} \textstyle\sum_{i=1}^{M} \boldsymbol{f}_i(\boldsymbol{x}) \tag{1}$$

Metrics for ensemble diversity. Two metrics of ensemble diversity are (1) variance [e.g. 39], and (2) Jensen-Shannon divergence [e.g. 45, 18]. Mathematically, they are (respectively) defined as:

$$\underset{p(\boldsymbol{f})}{\text{Var}}\left[\boldsymbol{f}(\boldsymbol{x})\right] = \textstyle\sum_{i=1}^{C} \underset{p(\boldsymbol{f})}{\text{Var}}\left[f^{(i)}(\boldsymbol{x})\right], \quad \underset{p(\boldsymbol{f})}{\text{JSD}}\left[y \,|\, \boldsymbol{f}(\boldsymbol{x})\right] = \text{H}\big[y \,|\, \bar{\boldsymbol{f}}(\boldsymbol{x})\big] - \underset{p(\boldsymbol{f})}{\mathbb{E}}\big[\text{H}\left[y \,|\, \boldsymbol{f}(\boldsymbol{x})\right]\big] \tag{2}$$

where $f^{(i)}$ refers to the probability assigned by a model to the i-th output class, and H is the entropy. Both metrics are always positive and minimized when the predictions from ensemble members are the same, i.e. not diverse.

Models and training datasets. We reuse and train a variety of neural networks on two benchmark image classification datasets: **CIFAR10** [41] and **ImageNet** [14]. In particular, we include the 137 CIFAR10 models trained by Miller et al. [52], corresponding to 32 different architectures each trained for 2-5 seeds; as well as the "standard" 78 ImageNet models curated by Taori et al. [72], each corresponding to a different architecture trained for 1 seed. To form homogeneous ensembles, we additionally train 10 network architectures on CIFAR10 and three on ImageNet. We train 5 independent instances of each model architecture, where each instance differs only in terms of initialization and minibatch ordering. We form homogeneous deep ensembles by combining 4 out of the 5 random seeds. From this process, we can consider 5 single model replicas and 5 ensemble replicas for each model architecture. Unless otherwise stated, ensembles are formed following Eq. (1).

[1] While it is also possible to average the logits (log probabilities) of each model, we note that probability averaging is far more common in the literature [e.g. 45].

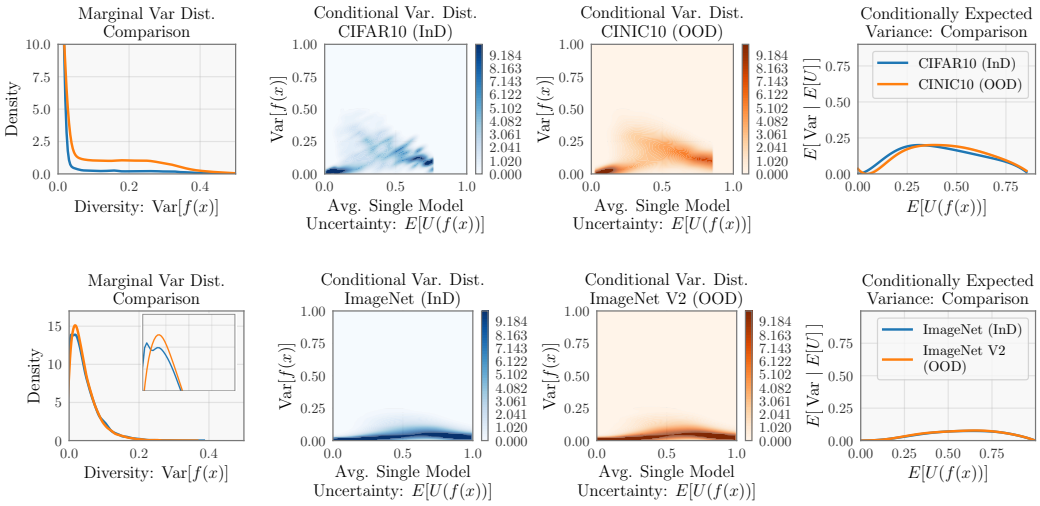

Figure 1: **Ensemble diversity does not yield better OOD uncertainty quantification, after controlling for average single model uncertainty.** Panels compare ensemble variance ($\mathrm{Var}[\boldsymbol{f}(\boldsymbol{x})]$) on InD (blue) vs. OOD (orange) data. The top row represents the variance for ensembles composed of 5 WideResNet 28-10 [78] networks evaluated on CIFAR10 and CINIC10, and the bottom row represents the variance for ensembles of 5 AlexNets, evaluated on ImageNet and ImageNetV2. The left column shows that, consistent with previous results, deep ensembles express higher variance predictions on OOD vs. InD data. The middle columns show $p(\mathrm{Var} \mid \mathbb{E}[U])$ (arguments suppressed for clarity) for InD (second column) and OOD data (third column). Suprisingly, we find that these conditional distributions are extremely similar. In the right columns, we further show the similarity of these conditional distributions (InD and OOD) using the conditional expectation $\mathbb{E}[\mathrm{Var} \mid \mathbb{E}[U]]$, estimated with kernel ridge regression. For experimental details, see Appx. F.

OOD datasets. A majority of our analysis compares deep ensembles on InD versus OOD test data. To that end, we consider three different catagories of OOD datasets as suggested by [52]: *Shifted reproduction datasets.* This category includes the **CIFAR10.1** and **ImageNetV2** datasets [66], both of which were collected and labeled following the same curation processes of the original CIFAR10 and ImageNet datasets, respectively. Neural networks (trained on the original datasets) tend to achieve worse performance on these new test sets. *Alternative benchmark datasets.* The **CINIC10** dataset [12] shares the same classes as CIFAR10 but uses images drawn and downsampled from the ImageNet dataset. Because ImageNet and CIFAR10 images were collected using different curation procedures, models trained on CIFAR10 tend to achieve worse performance on CINIC10. *Synthetically corrupted datasets.* The **CIFAR10C** and **ImageNetC** datasets [34], apply synthetic perturbations to CIFAR10 and ImageNet images (e.g. Gaussian blur, fog effects, etc.). Due to their synthetic nature, these datasets offer shifts of various intensity (e.g. mild blur versus heavy blur). We relegate most of our analysis of these datasets to the Appendix.

4 Hypothesis: ensemble diversity is responsible for improved UQ

The ability of deep ensembles to produce higher estimates of uncertainty on OOD data has been attributed to ensemble diversity [45, 18, 76]. In particular, ensemble diversity is hypothesized to increase on OOD data, where one would expect that OOD predictions from individual ensemble members are less constrained by their shared training data [45]. This hypothesis is attractive because it suggests that deep ensembles offer an additional mechanism for uncertainty quantification beyond what is afforded by any single model. In this section, we test this hypothesis by quantifying the contribution of ensemble diversity to a deep ensemble's total predictive uncertainty on both InD and OOD data.

4.1 Metrics for ensemble diversity

Common metrics for ensemble diversity provide interpretable decompositions of uncertainty: *ensemble uncertainty* = *ensemble diversity* + *average single model uncertainty*. For example, if we use variance (Eq. 2) as a metric for ensemble diversity [39, 27], then we show ensemble uncertainty can be decomposed as:

$$\overbrace{U\left(\bar{\boldsymbol{f}}(\boldsymbol{x})\right)}^{\text{ens. uncert.}} = \overbrace{\underset{p(\boldsymbol{f})}{\mathrm{Var}}\left[\boldsymbol{f}(\boldsymbol{x})\right]}^{\text{ens. diversity}} + \overbrace{\underset{p(\boldsymbol{f})}{\mathbb{E}}\left[U\left(\boldsymbol{f}(\boldsymbol{x})\right)\right]}^{\text{avg. single model uncert.}} . \tag{3}$$

where $\boldsymbol{f}(\boldsymbol{x}) \in \Delta^C$ is a probabilistic prediction, and $U(\boldsymbol{f}(\boldsymbol{x}))$ is a quadratic notion of uncertainty:

$$U\left(\boldsymbol{f}(\boldsymbol{x})\right) \triangleq 1 - \sum_{i=1}^{C}\left[p(y=i \mid \boldsymbol{f}(\boldsymbol{x}))\right]^2.$$

See derivation in Appx. C. Intuitively, U will be small when most probability is placed on a single class, and will be large when probability is distributed amongst classes. See Appx. C for analogous results with Jensen Shannon divergence as the diversity metric (Eq. 2). Based on our hypothesis, ensemble diversity (Var in Eq. 3) should increase on OOD data *independently* of average single model uncertainty ($\mathbb{E}[U]$). In other words, given *any* level of $\mathbb{E}[U]$, we would expect more ensemble diversity for OOD data than InD data.

4.2 Experiment: InD vs OOD ensemble diversity

We test 10 different ensembles of size $M = 5$ trained on CIFAR10, and three ensembles trained on ImageNet. We evaluate these ensembles on their respective InD (CIFAR10, Imagenet) and OOD (CIFAR10.1, CINIC10, CIFAR10C, ImageNet V2, ImageNetC) test sets. In Fig. 1, we analyze the variance of two of these deep ensembles, evaluated on CIFAR10 vs CINIC10 (top row) and ImageNet vs ImageNetV2 (bottom row), see Appx. F for a complete set of results. The left panel of Fig. 1 shows the distribution $p(\mathrm{Var})$ for InD and OOD data. Ensembles tend to express higher variance on OOD data than InD data; a finding consistent with previous work [45, 18]. However, we emphasize this result is not sufficient to directly attribute UQ improvements to ensemble diversity.

Controlling for single model uncertainty. A different picture emerges when we control for single model uncertainty. Fig. 1 (middle) shows histograms of $p(\mathrm{Var} \mid \mathbb{E}[U])$ i.e. the ensemble variance *conditioned on* average single model uncertainty as given by Eq. (3). Surprisingly, we see that the OOD and InD conditional distributions are very similar. We further study this similarity in Fig. 1 (right), which plots expected ensemble variance conditioned on average single model uncertainty: $\mathbb{E}[\,\mathrm{Var} \mid \mathbb{E}[U]\,]$. Far from what our hypothesis would suggest (i.e. higher OOD diversity across all levels of average single model uncertainty) we observe that the conditional expectation of ensemble diversity on InD vs OOD data is nearly identical. In Appx. F (Fig. 8-Fig. 12), we offer statistical validation of these observations, and further demonstrate that this phenomenon holds across various architectures, InD, and OOD datasets. In all cases, the difference between the InD and OOD expected variance is only a few percentage points, and/or not statistically significant.

Understanding the relationship between ensemble diversity and average single model uncertainty. By controlling for average single model uncertainty, we see that ensemble diversity does not differ significantly for InD versus OOD data. In turn, these results imply that the InD/OOD difference we see in Fig. 1 (left) must be due entirely to a change in the distribution of *average single model uncertainty*, $p(\mathbb{E}[U])$. From these results, we can conclude that surprisingly, the UQ benefits of ensemble diversity are dictated by the corresponding average single model uncertainty. In Appx. F.1 we plot the differences in $p(\mathbb{E}[U])$ that drive the changes in ensemble diversity observed in Fig. 1 (left).

4.3 What does ensemble diversity actually measure?

Our analysis above shows that ensemble diversity is not directly responsible for the improved OOD uncertainty estimates offered by ensembles. To begin to understand why this might be the case, it is useful to consider the link between ensemble diversity and performance. It has long been established that diversity amongst ensemble members is a necessary and sufficient condition for the superior performance of ensembles [e.g. 16]. To demonstrate this, consider any strictly convex loss function,

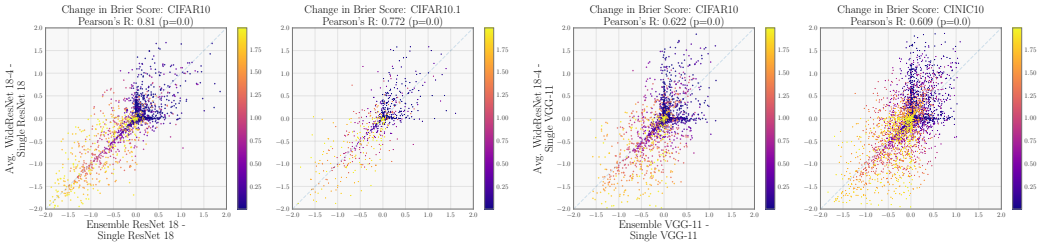

Figure 2: **Ensemble diversity is meaningfully correlated with the expected improvements from increasing model capacity**. (Left) Panels illustrate the per-datapoint gains in Brier score over a single ResNet 18 model by either forming a deep ensemble of ResNet 18 models (x-axis), or by increasing single model capacity, here with a WideResNet 18-4 (y-axis). The ResNet-18 ensemble and WideResNet 18-4 achieve nearly identical performance and strongly correlated improvements on both CIFAR10 and CIFAR10.1. Colors indicate the Brier score achieved by the single ResNet 18 model on each datapoint. (Right) We repeat the experiment for CIFAR10/CINIC10, showing the gains in Brier score over a VGG11 model, using either an ensemble of VGG11, or a WideResNet 18-4 model. Improvements are indistinguishable from relevant controls, and corresponding model accuracies are well matched, as shown in Appx. I.

such as negative log likelihood (NLL) or the multiclass Brier score (B) [8]:

$$\mathrm{NLL}(\boldsymbol{f}(\boldsymbol{x}), y) \triangleq -\log\left(f^{(y)}(\boldsymbol{x})\right), \quad \mathrm{B}(\boldsymbol{f}(\boldsymbol{x}), y) \triangleq \|\boldsymbol{f}(\boldsymbol{x}) - \mathbf{1}_y\|_2^2. \tag{4}$$

(Here, $\mathbf{1}_y$ represents a one-hot encoding of y.) Recall that the ensemble prediction $\bar{\boldsymbol{f}}(\boldsymbol{x})$ is the average of all model predictions (i.e. $\mathbb{E}_{p(\boldsymbol{f})}[\boldsymbol{f}(\boldsymbol{x})]$). By Jensen's inequality:

$$\mathrm{NLL}(\bar{\boldsymbol{f}}(\boldsymbol{x}), y) \leq \mathop{\mathbb{E}}_{p(\boldsymbol{f})}\left[\mathrm{NLL}(\boldsymbol{f}(\boldsymbol{x}), y)\right], \quad \mathrm{B}(\bar{\boldsymbol{f}}(\boldsymbol{x}), y) \leq \mathop{\mathbb{E}}_{p(\boldsymbol{f})}\left[\mathrm{B}(\boldsymbol{f}(\boldsymbol{x}), y)\right] \tag{5}$$

In other words, the performance of the ensemble (as measured by NLL or Brier score) must be better than the average performance of ensemble members. Because both NLL and Brier score are strictly convex, the Jensen gap in Eq. (5) will grow as $p(\boldsymbol{f})$ becomes less constant, or more "diverse." In particular, the Jensen gap for Brier score is exactly equal to the ensemble variance (Eq. 2):

$$\mathrm{B}(\bar{\boldsymbol{f}}(\boldsymbol{x}), y) - \mathop{\mathbb{E}}_{p(\boldsymbol{f})}\left[\mathrm{B}(\boldsymbol{f}(\boldsymbol{x}), y)\right] = \mathop{\mathrm{Var}}_{p(\boldsymbol{f})}[\boldsymbol{f}(\boldsymbol{x})]. \tag{6}$$

(Similar results are well known in the regression context—[e.g. 42, 51]—see Appx. D for a short derivation). In other words, $\mathrm{Var}_{p(\boldsymbol{f})}[\boldsymbol{f}(\boldsymbol{x})]$ measures the expected predictive improvement we obtain through ensembling. We can use these results to investigate our UQ findings. Hypothetically, if $\mathrm{Var}_{p(\boldsymbol{f})}[\boldsymbol{f}(\boldsymbol{x})]$ were also responsible for improving UQ, this would imply that the performance gains from ensembling are somehow fundamentally different than the performance gains from increasing a single model's capacity, as the latter can hurt uncertainty estimates [26]. However, in the next section we demonstrate that these two methods of increasing performance are in fact correlated.

4.4 Ensembling versus increasing model capacity

In Fig. 2, we compare the expected per-datapoint performance improvement gained through ensembling (x-axis) to the performance improvement gained through increasing model capacity (y-axis). Specifically, we compare an ensemble of 4 CIFAR10 models (ResNet18) with a single large model (WideResNet-18-4). The ensemble and the large single model achieve comparable Brier Score: 0.084 ± 0.002 on the InD test dataset and 0.210 ± 0.002 on the CIFAR10.1 OOD dataset. In Fig. 2 (left), we plot the Brier score of the ensemble versus the large model on a per-datapoint level, depicting the *improvement correlation* across the dataset.

Surprisingly, we find that increasing model capacity and ensembling yield very similar performance improvements *on most datapoints*. The ensemble improvements and large model improvements have a Pearson's correlation of 0.81 on the InD test set. Importantly, we see that this correlation is preserved even on OOD data (Pearson's correlation: 0.76). We replicate this result for a different

ensemble/larger model pair (VGG-11 ensemble versus WideResNet-18-4) that again have nearly identical InD and OOD performance: 0.093 ± 0.004 CINIC10 InD Brier Score; 0.48 ± 0.02 CINIC10 OOD Brier Score (Fig. 2, right). We compare each improvement correlation in Fig. 2 to relevant controls, and ensure comparable accuracies (Appx. I). In all cases we find that improvements are as similar as we might expect if comparing two performance matched ensembles, or two single models. This result is unexpected, because the ensemble and the large model represent two distinct architectures (ResNet versus WideResNet) and two different modes of training (independent training of separate models versus training one large model). Recalling the relationship between ensemble diversity and relative performance gains, these results suggest that *ensemble diversity estimates the improvement we should expect by increasing model capacity.* We conclude that, with regards to UQ and performance improvements, ensemble diversity offers no significant benefit over what can be obtained with single models.

4.5 Implications for uncertainty estimation

Epistemic vs. aleatoric uncertainty. Uncertainty is often categorized as coming from one of two components [e.g. 38]. The *epistemic* component is said to capture uncertainty due to a limited number of observations, or uncertainty that the model accurately and uniquely captures the ground truth labeling process. Apparently, it can be reduced by collecting more data. In contrast, the *aleatoric* component is described as capturing the inherent ambiguity in the data (e.g. a blurry image) and is considered to be irreducible noise. In decision making applications such as active learning [68, 22] or model-based reinforcement learning [44, 77], this uncertainty decomposition is employed to identify informative datapoints for our model to sample next [15]. Previous work has interpreted ensemble diversity as in Eq. (2) as epistemic uncertainty [50, 27, 77], with average single model uncertainty in Eqs. (3) and (7) identified as aleatoric uncertainty correspondingly [70]. Our results in Fig. 1 demonstrate that there is a limitation to this interpretation, as we would expect more ensemble variance (the proxy for epistemic uncertainty) for OOD data than for InD data, independent of single model uncertainty (the proxy for aleatoric uncertainty). We therefore suggest caution when using ensembles to differentiate sources of uncertainty in downstream applications.

Bayesian perspective. Bayesian model averaging, or BMA integrates predictions against a posterior distribution over models. Given training data $\mathcal{D}$, BMA forms the prediction $p(y \mid \boldsymbol{x}, \mathcal{D}) = \int \boldsymbol{f}(\boldsymbol{x})\, p(\boldsymbol{f} \mid \mathcal{D})\, \mathrm{d}\boldsymbol{f}$. The advantage of BMA is the ability to consider all possible predictions given a prior and conditioned on training data, thereby mitigating the risk in estimating the "true" model from limited data. A recent line of work argues that modern deep ensembles (unlike classic ensembles—see Minka [53]) can be viewed as approximate BMA [35, 76], although we also note that concurrent work emphasizes differences between deep ensembles and Bayesian inference in the infinite width limit [31]. Our results in Fig. 1 identify a limitation of ensembles as approximate Bayesian inference. The posterior predictive distribution should express higher variance for OOD data than InD data, which is not the case for the deep ensemble predictive distribution. In Appx. E, we demonstrate that exact Bayesian inference does yield higher OOD posterior variance, even after conditioning on observational noise. We emphasize that our results neither agree nor disagree with the BMA interpretation of ensembling. Rather they suggest that ensemble members should not be interpreted as true posterior samples, and that (as with many approximate Bayesian methods) the ensemble approximation to BMA is biased.

5 Hypothesis: ensemble diversity is responsible for improved robustness

Beyond uncertainty quantification, ensembles have been shown to often achieve better predictive performance than single networks (as measured by 0-1 accuracy, NLL, or Brier score) on OOD or shifted datasets [45, 62, 27]. In this section, we test the hypothesis that ensemble diversity improves robustness over what single neural networks can offer.

5.1 Effective robustness

We use the concept of "effective robustness" as introduced by Taori et al. [72]. These authors note that there is often a deterministic relationship between a neural network's accuracy on InD data and its accuracy on an OOD dataset (green line in Fig. 3). In other words, any improvements in OOD performance can be entirely explained by improvements in OOD performance. A model is

considered to be *effectively robust* only if it achieves better OOD accuracy than what is predicted by its InD accuracy. In general, there are very few neural networks or training procedures that exhibit effective robustness against any OOD dataset [72, 52]. To measure the role that ensemble diversity plays in robustness, we quantify to what extent deep ensemble OOD performance can be explained by InD performance (as measured by the deterministic relationship derived from single models). If the performance of deep ensembles follows the same deterministic relationship, then deep ensembles are not effectively robust (i.e. multiple diverse predictors offer no additional robustness over what a single neural network provides).

5.2 Experiment: measuring effective robustness of deep ensembles across metrics

Ensembles are not effectively robust with respect to 0-1 accuracy. Following Miller et al. [52], we measure the InD and OOD error for all the models described in Sec. 3. The top left of Fig. 4 compares the error of models on CIFAR10 (InD) versus CINIC10 (OOD), and the bottom left plot compares the error of models on ImageNet (InD) versus ImageNetV2 (OOD). From these plots, we observe several trends. In agreement with Taori et al. [72] and Miller et al. [52], we observe that single models (green dots) follow a colinear relationship for InD versus OOD accuracy. Additionally, we find that *ensembles (orange dots) do not deviate from this colinear InD/OOD relationship.* In Appx. H.1, we evaluate the quality of these linear trends. In particular, we fit separate linear trend lines for individual models and deep ensembles. All trend lines achieve correlations of $R > 0.84$, and their coefficients only differ by 1% at most. This suggests that, after controlling for InD accuracy, the OOD accuracy of ensembles is nearly identical to that expected of single models. (See Appx. H for CIFAR10.1/CIFAR10C/ImageNetC results.)

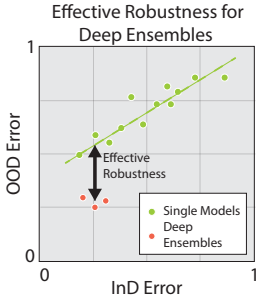

Figure 3: **Cartoon of effectively robust deep ensembles (what we want).**

Ensembles are not effectively robust with respect to NLL or Brier score. Although deep ensembles are not effectively robust in terms of predictive accuracy, many of their robustness benefits have been reported in terms of probabilistic metrics, such as NLL or Brier score [62]. We therefore extend our investigation of deep ensemble effective robustness to these metrics. Fig. 4 (middle left) plots the InD NLL and OOD NLL of various ensembles and single models. To the best of our knowledge, this is the first time that the effective robustness experiments of Taori et al. [72] and Miller et al. [52] have been extended to metrics other than 0-1 accuracy. We observe that the relationship between InD NLL and OOD NLL is not as linear as the accuracy trend. Nevertheless, we observe no discernible difference between the performance of single networks and ensembles (see Appx. H.1 for a quantitative analysis). We observe a similiar phenomenon when we plot InD versus OOD Brier score (Fig. 4, middle right)—ensembles and single models obtain similar OOD Brier score, after controlling for InD Brier score. Our key conclusion is that deep ensembles fail to demonstrate effective robustness when evaluated on probabilistic performance metrics, just as they do with 0-1 accuracy. (See Appx. H for CIFAR10.1/CIFAR10C/ImageNetC results.)

Ensembles do not offer effectively robust calibration. We also compare InD and OOD calibration for various single models and ensembles. We consider various metrics for measuring and comparing calibration used throughout the literature. Expected Calibration Error (ECE) [58] is a standard metric for measuring calibration of neural networks. As we show in Appx. H.3, there is little correlation between a single model's InD ECE and OOD ECE, which precludes any discussion of "effective robustness" using this metric. Conversely, Fig. 4 depicts a strong correlation between a model's InD/OOD square root of the Expected *Squared* Calibration Error (rESCE) [13, 56], which appears in a common decomposition of the Brier score [9]. We therefore expect that any InD/OOD trend for the rESCE should be qualitatively similar to the InD/OOD trends observed for Brier score. In Fig. 4 (top right), we observe a linear trend relating the CIFAR10 (InD) and CINIC10 (OOD) rESCE of single models. The rESCE of the ImageNet models, Fig. 4 (bottom right), follows a bimodal trend, where—depending on the model architecture—InD rESCE is correlated with either low or high OOD calibration. Nevertheless, for both datasets we find that ensembles do not achieve better OOD calibration that single models with similar InD calibration. (See Appx. H for CIFAR10.1/CIFAR10C/ImageNetC results.)

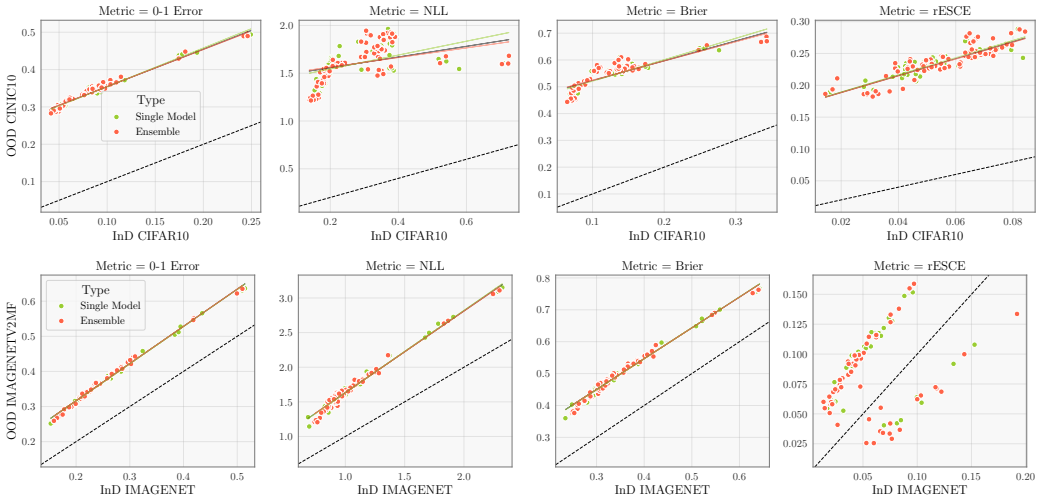

Figure 4: **Deep ensembles are not "effectively robust" across a variety of performance metrics**. Panels illustrate InD vs OOD performance metrics, from left to right: 0-1 Error, NLL, Brier Score, and rESCE. The model types considered are single models and ensembles. Linear trend lines are shown in solid lines, and black dotted lines indicate perfect robustness. We find that, conditioned on InD performance, ensembles offer no better OOD performance than single models. See Appx. H for additional corruptions.

5.3 Heterogeneous and implicit ensembles

From the previous results, it is clear that—by many metrics—ensembling multiple copies of the same model architecture confers no additional robustness over single models. A natural question is whether we could achieve more robustness by ensembling different model architectures together. To test this hypothesis, we repeat the same robustness experiments with *heterogeneous ensembles*: ensembles that combine multiple architectures, and *implicit ensembles*: single models that approximate deep ensembles, usually through parameter sampling [21]. To construct heterogeneous ensembles, we divide the 137 CIFAR10 models and 78 ImageNet models from Sec. 3 based on their InD accuracy. Ensembles are then formed by randomly selecting 4 models from each bin. This procedure ensures that all ensemble members will have similar accuracy, even though the ensemble members may represent different architectures and training regimens. Despite their additional diversity, these heterogeneous ensembles do not provide effective robustness, as shown in Appx. H.4. Finally, we investigate if these results also follow for three implicit ensembling mechanisms: Monte Carlo Dropout [21], multiple-input-multiple-output (MIMO) [30], and Batch Ensembles [75]. We find that implicit ensembles are also not effectively robust, as depicted in Appx. H.4.

5.4 Implications.

As discussed in Sec. 4.3, ensemble diversity is responsible for improved NLL and Brier score relative to constituent models. In this sense, ensemble diversity is responsible for improved OOD performance. However, these OOD improvements exactly follow the deterministic trends predicted by (standard) single models, and thus ensembling multiple diverse predictors does not yield any "effective robustness" over what could be achieved by a better performing single model. Unlike prior research [e.g. 62, 27], these results suggest that ensembles are a tool of convenience for obtaining better OOD performance, but not qualitatively different from single models in this respect.

6 Discussion

In this work, we rigorously test common intuitions about the benefits of deep ensembles to UQ and robustness, and find these explanations wanting. Below, we lay out limitations of our study, summarize our conclusions, and indicate important lines of future work.

Ensembling in the overparametrized regime. We emphasize that our analysis only focuses on ensembles of neural networks, and does not necessarily apply to ensembling techniques in general (e.g. random forests or gradient boosted decision trees). Indeed, we predict that many of our results are direct consequences of the fact that we are ensembling high-capacity "interpolating" models, which seem to generalize well despite being massively overparametrized [5, 1, 59, 29]. In future work, we will examine the effect of overparametrization directly by replicating these experiments with ensembles of weak learners.

Neural network uncertainty quantification. In examining the conditional distributions in Fig. 1, we see that OOD uncertainty quantification is not directly impacted by ensemble diversity. These findings show that the role of ensemble diversity in deep ensemble UQ is far more limited than previously hypothesized [e.g. 45, 18, 27].

Effective robustness. Our results in Figure 4 show that ensemble diversity does not yield improvements to robustness that cannot be explained by InD performance. This finding is in line with other results demonstrating that effective robustness is very difficult to achieve [2].

When should we use deep ensembles? Despite our results, we maintain that ensembling can be viewed as a reliable "black box" method of improving neural network performance, both InD and OOD. It is simple (though potentially expensive) to improve upon a model through ensembling, and training a single model that matches the performance of an ensemble is not always straightforward [40, 48, 74]. However we caution that deep ensembles are not a panacea for the issues faced by single models. In particular, it is dangerous to assume that deep ensembles mitigate the robustness issues of single models in contexts where we can expect dataset shift, or that ensemble diversity provides a reliable baseline for model uncertainty in the absence of ground truth. Thus, for many practitioners, the choice of using a deep ensemble versus a performance matched single model may ultimately be dictated by practical considerations, such as performance given a pre-determined parameter/FLOP budget for model training and evaluation [40, 48, 74]. Beyond these practical concerns, we have yet to find evidence for any reason to prefer the use of deep ensembles over an appropriately chosen single model.

Acknowledgments and Disclosure of Funding

We thank John Miller for sharing models trained on CIFAR10, and Taori et al. [72] for making their trained ImageNet models and code open sourced and easy to use. We would also like to thank Dustin Tran for his insightful comments, and Julien Boussard for helpful discussions on statistical testing. TA is supported by NIH training grant 2T32NS064929-11. EKB is supported by NIH 5T32NS064929-13, NSF 1707398, and Gatsby Charitable Foundation GAT3708. GP and JPC are supported by the Simons Foundation, McKnight Foundation, Grossman Center for the Statistics of Mind, and Gatsby Charitable Trust.

References

[1] Ben Adlam and Jeffrey Pennington. Understanding double descent requires a fine-grained bias-variance decomposition. *Advances in neural information processing systems*, 33:11022–11032, 2020.

[2] Anders Andreassen, Yasaman Bahri, Behnam Neyshabur, and Rebecca Roelofs. The evolution of out-of-distribution robustness throughout fine-tuning. *arXiv preprint arXiv:2106.15831*, 2021.

[3] Luis Antonio Ortega Andrés, Rafael Cabañas, and Andres Masegosa. Diversity and generalization in neural network ensembles. In *International Conference on Artificial Intelligence and Statistics*, pages 11720–11743. PMLR, 2022.

[4] Arsenii Ashukha, Alexander Lyzhov, Dmitry Molchanov, and Dmitry Vetrov. Pitfalls of in-domain uncertainty estimation and ensembling in deep learning. *arXiv preprint arXiv:2002.06470*, 2020.

[5] Mikhail Belkin, Daniel Hsu, Siyuan Ma, and Soumik Mandal. Reconciling modern machine-learning practice and the classical bias–variance trade-off. *Proceedings of the National Academy of Sciences*, 116(32):15849–15854, 2019.

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

[7] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[8] Glenn W Brier et al. Verification of forecasts expressed in terms of probability. *Monthly weather review*, 78(1):1–3, 1950.

[9] Jochen Bröcker. Reliability, sufficiency, and the decomposition of proper scores. *Quarterly Journal of the Royal Meteorological Society: A journal of the atmospheric sciences, applied meteorology and physical oceanography*, 135(643):1512–1519, 2009.

[10] Rich Caruana, Alexandru Niculescu-Mizil, Geoff Crew, and Alex Ksikes. Ensemble selection from libraries of models. In *Proceedings of the twenty-first international conference on Machine learning*, page 18, 2004.

[11] Kamil Ciosek, Vincent Fortuin, Ryota Tomioka, Katja Hofmann, and Richard Turner. Conservative uncertainty estimation by fitting prior networks. In *International Conference on Learning Representations*, 2019.

[12] Luke N Darlow, Elliot J Crowley, Antreas Antoniou, and Amos J Storkey. CINIC-10 is not ImageNet or CIFAR-10. *arXiv preprint arXiv:1810.03505*, 2018.

[13] Morris H DeGroot and Stephen E Fienberg. The comparison and evaluation of forecasters. *Journal of the Royal Statistical Society: Series D (The Statistician)*, 32(1-2):12–22, 1983.

[14] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition*, pages 248–255, 2009.

[15] Stefan Depeweg, Jose-Miguel Hernandez-Lobato, Finale Doshi-Velez, and Steffen Udluft. Decomposition of uncertainty in bayesian deep learning for efficient and risk-sensitive learning. In *International Conference on Machine Learning*, pages 1184–1193. PMLR, 2018.

[16] Thomas G Dietterich. Ensemble methods in machine learning. In *International Workshop on Multiple Classifier Systems*, pages 1–15, 2000.

[17] Pedro M. Domingos. Why does bagging work? a Bayesian account and its implications. In *KDD*, 1997.

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

[19] Yoav Freund. Boosting a weak learning algorithm by majority. *Information and computation*, 121(2):256–285, 1995.

[20] Jerome H Friedman. Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pages 1189–1232, 2001.

[21] Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pages 1050–1059. PMLR, 2016.

[22] Yarin Gal, Riashat Islam, and Zoubin Ghahramani. Deep bayesian active learning with image data. In *International Conference on Machine Learning*, 2017.

[23] Jacob R Gardner, Geoff Pleiss, David Bindel, Kilian Q Weinberger, and Andrew Gordon Wilson. Gpytorch: Blackbox matrix-matrix gaussian process inference with gpu acceleration. *arXiv preprint arXiv:1809.11165*, 2018.

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

[25] Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *The Journal of Machine Learning Research*, 13(1):723–773, 2012.

[26] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *International Conference on Machine Learning*, 2017.

[27] Fredrik K Gustafsson, Martin Danelljan, and Thomas B Schon. Evaluating scalable bayesian deep learning methods for robust computer vision. In *Computer Vision and Pattern Recognition Workshops*, pages 318–319, 2020.

[28] Lars Kai Hansen and Peter Salamon. Neural network ensembles. *Transactions on pattern analysis and machine intelligence*, 12(10):993–1001, 1990.

[29] Trevor Hastie, Andrea Montanari, Saharon Rosset, and Ryan J Tibshirani. Surprises in high-dimensional ridgeless least squares interpolation. *The Annals of Statistics*, 50(2):949–986, 2022.

[30] Marton Havasi, Rodolphe Jenatton, Stanislav Fort, Jeremiah Zhe Liu, Jasper Snoek, Balaji Lakshminarayanan, Andrew Mingbo Dai, and Dustin Tran. Training independent subnetworks for robust prediction. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=OGg9XnKxFAH.

[31] Bobby He, Balaji Lakshminarayanan, and Yee Whye Teh. Bayesian deep ensembles via the neural tangent kernel. In *Advances in Neural Information Processing Systems*, 2020.

[32] Bobby He, Balaji Lakshminarayanan, and Yee Whye Teh. Bayesian deep ensembles via the neural tangent kernel. *Advances in neural information processing systems*, 33:1010–1022, 2020.

[33] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[34] Dan Hendrycks and Thomas G Dietterich. Benchmarking neural network robustness to common corruptions and surface variations. In *International Conference on Learning Representations*, 2019.

[35] Lara Hoffmann and Clemens Elster. Deep ensembles from a bayesian perspective. *arXiv preprint arXiv:2105.13283*, 2021.

[36] Gao Huang, Yixuan Li, Geoff Pleiss, Zhuang Liu, John E Hopcroft, and Kilian Q Weinberger. Snapshot ensembles: Train 1, get m for free. *International Conference on Learning Representations*, 2017.

[37] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017.

[38] Eyke Hüllermeier and Willem Waegeman. Aleatoric and epistemic uncertainty in machine learning: An introduction to concepts and methods. *Machine Learning*, 110(3):457–506, 2021.

[39] Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision? In *Advances in Neural Information Processing Systems*, 2017.

[40] Dan Kondratyuk, Mingxing Tan, Matthew Brown, and Boqing Gong. When ensembling smaller models is more efficient than single large models. *arXiv preprint arXiv:2005.00570*, 2020.

[41] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images, 2009.

[42] Vedelsby Neural Network Ensembles Krogh. Cross validation and active learning advances in neural information processing systems 7, 1995.

[43] Ananya Kumar, Aditi Raghunathan, Tengyu Ma, and Percy Liang. Calibrated ensembles: A simple way to mitigate ID-OOD accuracy tradeoffs. In *NeurIPS 2021 Workshop on Distribution Shifts: Connecting Methods and Applications*, 2021.

[44] Thanard Kurutach, Ignasi Clavera, Yan Duan, Aviv Tamar, and Pieter Abbeel. Model-ensemble trust-region policy optimization. In *International Conference on Learning Representations*, 2018.

[45] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in Neural Information Processing Systems*, 2017.

[46] 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.

[47] Jeremiah Liu, Zi Lin, Shreyas Padhy, Dustin Tran, Tania Bedrax Weiss, and Balaji Lakshminarayanan. Simple and principled uncertainty estimation with deterministic deep learning via distance awareness. *Advances in Neural Information Processing Systems*, 33:7498–7512, 2020.

[48] Ekaterina Lobacheva, Nadezhda Chirkova, Maxim Kodryan, and Dmitry P Vetrov. On power laws in deep ensembles. In *Advances in Neural Information Processing Systems*, 2020.

[49] Wesley J Maddox, Pavel Izmailov, Timur Garipov, Dmitry P Vetrov, and Andrew Gordon Wilson. A simple baseline for bayesian uncertainty in deep learning. In *Advances in Neural Information Processing Systems*, 2019.

[50] Andrey Malinin and Mark Gales. Predictive uncertainty estimation via prior networks. *Advances in neural information processing systems*, 31, 2018.

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

[52] John P Miller, Rohan Taori, Aditi Raghunathan, Shiori Sagawa, Pang Wei Koh, Vaishaal Shankar, Percy Liang, Yair Carmon, and Ludwig Schmidt. Accuracy on the line: on the strong correlation between out-of-distribution and in-distribution generalization. In *International Conference on Machine Learning*, 2021.

[53] Thomas P Minka. Bayesian model averaging is not model combination. *Available electronically at http://www. stat. cmu. edu/minka/papers/bma. html*, pages 1–2, 2000.

[54] Mohammad Moghimi, Serge J Belongie, Mohammad J Saberian, Jian Yang, Nuno Vasconcelos, and Li-Jia Li. Boosted convolutional neural networks. In *British Machine Vision Conference*, 2016.

[55] Paul W Munro and Bambang Parmanto. Competition among networks improves committee performance. In *Advances in Neural Information Processing Systems*, 1997.

[56] Allan H Murphy and Robert L Winkler. Reliability of subjective probability forecasts of precipitation and temperature. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 26(1):41–47, 1977.

[57] Zachary Nado, Neil Band, Mark Collier, Josip Djolonga, Michael Dusenberry, Sebastian Farquhar, Angelos Filos, Marton Havasi, Rodolphe Jenatton, Ghassen Jerfel, Jeremiah Liu, Zelda Mariet, Jeremy Nixon, Shreyas Padhy, Jie Ren, Tim Rudner, Yeming Wen, Florian Wenzel, Kevin Murphy, D. Sculley, Balaji Lakshminarayanan, Jasper Snoek, Yarin Gal, and Dustin Tran. Uncertainty Baselines: Benchmarks for uncertainty & robustness in deep learning. *arXiv preprint arXiv:2106.04015*, 2021.

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

[59] Preetum Nakkiran, Gal Kaplun, Yamini Bansal, Tristan Yang, Boaz Barak, and Ilya Sutskever. Deep double descent: Where bigger models and more data hurt. *Journal of Statistical Mechanics: Theory and Experiment*, 2021(12):124003, 2021.

[60] David Opitz and Richard Maclin. Popular ensemble methods: An empirical study. *Journal of Artificial Intelligence Research*, 11:169–198, 1999.

[61] Ian Osband, Zheng Wen, Mohammad Asghari, Morteza Ibrahimi, Xiyuan Lu, and Benjamin Van Roy. Epistemic neural networks. *arXiv preprint arXiv:2107.08924*, 2021.

[62] Yaniv Ovadia, Emily Fertig, Jie Ren, Zachary Nado, David 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 *Advances in Neural Information Processing Systems*, 2019.

[63] Michael P Perrone and Leon N Cooper. When networks disagree: Ensemble methods for hybrid neural networks, 1992.

[64] Rahul Rahaman and Alexandre H Thiery. Uncertainty quantification and deep ensembles. In *Advances in Neural Information Processing Systems*, 2021.

[65] Carl Edward Rasmussen and Christopher K Williams. *Gaussian processes for machine learning*. MIT press Cambridge, MA, 2006.

[66] Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do ImageNet classifiers generalize to ImageNet? In *International Conference on Machine Learning*, 2019.

[67] Robert E Schapire. The strength of weak learnability. *Machine learning*, 5(2):197–227, 1990.

[68] Burr Settles. Active learning literature survey, 2009.

[69] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[70] Lewis Smith and Yarin Gal. Understanding measures of uncertainty for adversarial example detection. *Conference on Uncertainty in Artificial Intelligence*, 2018.

[71] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Computer Vision and Pattern Recognition*, 2015.

[72] Rohan Taori, Achal Dave, Vaishaal Shankar, Nicholas Carlini, Benjamin Recht, and Ludwig Schmidt. Measuring robustness to natural distribution shifts in image classification. In *Advances in Neural Information Processing Systems*, 2020.

[73] Dustin Tran, Jeremiah Liu, Michael W Dusenberry, Du Phan, Mark Collier, Jie Ren, Kehang Han, Zi Wang, Zelda Mariet, Huiyi Hu, et al. Plex: Towards reliability using pretrained large model extensions. *arXiv preprint arXiv:2207.07411*, 2022.

[74] Abdul Wasay and Stratos Idreos. More or less: When and how to build convolutional neural network ensembles. In *International Conference on Learning Representations*, 2020.

[75] Yeming Wen, Dustin Tran, and Jimmy Ba. BatchEnsemble: an alternative approach to efficient ensemble and lifelong learning. In *International Conference on Learning Representations*, 2020.

[76] Andrew Gordon Wilson and Pavel Izmailov. Bayesian deep learning and a probabilistic perspective of generalization. In *Advances in Neural Information Processing Systems*, 2020.

[77] Tianhe Yu, Garrett Thomas, Lantao Yu, Stefano Ermon, James Zou, Sergey Levine, Chelsea Finn, and Tengyu Ma. MOPO: Model-based offline policy optimization. In *Advances in Neural Information Processing Systems*, 2020.

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

[79] Sheheryar Zaidi, Arber Zela, Thomas Elsken, Chris Holmes, Frank Hutter, and Yee Whye Teh. Neural ensemble search for uncertainty estimation and dataset shift. In *Advances in Neural Information Processing Systems*, 2021.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes] We relate our claims to two hypotheses in the introduction Sec. 1, and test each hypothesis in the corresponding results sections Secs. 4 and 5.
   (b) Did you describe the limitations of your work? [Yes] We specify in the discussion Sec. 6 that our results are limited to neural network ensembles, as opposed to more general ensembles.
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] We do so in Appx. A
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes]
   (b) Did you include complete proofs of all theoretical results? [Yes]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] We provide a link to a repository in the supplemental material section Appx. B. This repository contains instructions to reproduce main figures and to download relevant data.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] We provide instructions in the code Appx. B which specify internally the data splits and hyperparameters we used. We further specify in Appx. G that we chose default hyperparameters as specified in a separate code repo.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] In accordance with checklist guidelines, we report the fact that we ran statistical significance tests for our main results here- in particular, Appx. F describes tests for Fig. 1 and related results, Appx. I describes tests for Fig. 2 and related results, and Appx. H describes tests for Fig. 4 and related results.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] We do so in Appx. B
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] In Sec. 3, we reference the origin of the models [52] and [72], and provide further details in Appx. B.
   (b) Did you mention the license of the assets? [No] We provide links to publicly released assets with relevant licenses, but do not have a license for models that we were provided by the authors of [52].
   (c) Did you include any new assets either in the supplemental material or as a URL? [N/A] We do not provide new assets.
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [Yes] In our acknowledgements we thank the authors of [52] for agreeing to share their data with us- all other data is released under a pubic license.
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A] We do not believe this to apply to our data, which consists of deep network models trained using popular deep learning frameworks on benchmark datasets.
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

A Societal impact

Deep ensembles are popular in many real world applications, and a potential negative impact of our work is to expose flaws in applications reliant upon deep ensembles, especially in adversarial settings like fraud detection (although this may lead to improved systems further on as well).

B Code, data and compute

B.1 Code and data

We provide general directions to reproduce the main results of our paper in the linked directions here: https://github.com/cellistigs/interp_ensembles#readme.

These directions reference two repositories, corresponding to two separate branches of our codebase. The "compare_performance" branch can be found here: https://github.com/cellistigs/interp_ensembles/tree/compare_performance. Likewise, the "imagenet_pl" branch can be found here: https://github.com/cellistigs/interp_ensembles/tree/imagenet_pl. This code repository, together with the instructions provided above, specify all training and visualization details relevant to our study.

Finally, we share relevant data as a Zenodo repository: https://zenodo.org/record/6582653#.Yo7R0y-B3fZ. This data provides the logit outputs from individual models (and some ensembles) on the in and out of distribution data that we consider. These data are referenced in the code above.

B.2 Compute

We ran all CIFAR10 model training on Amazon Web Services (AWS), using the "p3.2xlarge" instance type with a Tesla V100 GPU. We ran half of ImageNet model training on an internal cluster with GeForce RTX 2080 Ti GPUs, and the other half on AWS with the "p3.8xlarge" instance type, again with Tesla V100 GPUs. Visualization and statistical testing was run on M1 MacBook Airs, and additionally on AWS "p3.2xlarge" and "p3.8xlarge" instances when additional capacity was required.

We show results for 50 models trained on CIFAR10, and 15 models trained on ImageNet. We estimate that on average, our CIFAR10 models required 3 hours of compute to train, and our ImageNet models required 48 hours to train. Finally, we estimate an additional 8 hours of compute required to run statistical tests and visualize results, resulting in a total of approximately 878 hours of total compute.

C Decompositions for uncertainty metrics

C.1 Jensen-Shannon divergence and entropic uncertainty

If we use Jensen-Shannon divergence (Eq. 2) as a metric for ensemble diversity [45, 18], we show ensemble uncertainty can be decomposed as:

$$\overbrace{\mathrm{H}\left[y \,|\, \bar{\boldsymbol{f}}(\boldsymbol{x})\right]}^{\text{ens. uncert.}} = \overbrace{\underset{p(\boldsymbol{f})}{\mathrm{JSD}}\left[y \,|\, \boldsymbol{f}(\boldsymbol{x})\right]}^{\text{ens. diversity}} + \overbrace{\underset{p(\boldsymbol{f})}{\mathbb{E}}\left[\mathrm{H}\left[y \,|\, \boldsymbol{f}(\boldsymbol{x})\right]\right]}^{\text{avg. single model uncert.}}. \tag{7}$$

where $\mathrm{H}[y \,|\, \cdot]$ represents the entropy of a categorical distribution parameterized by $(\cdot)$.

Furthermore, $\mathrm{JSD}_{p(\boldsymbol{f})}\left[y \,|\, \boldsymbol{f}(\boldsymbol{x})\right] = \frac{1}{M}\sum_{m=1}^{M} KL[y \mid \boldsymbol{f}(\boldsymbol{x}) \| y \mid \bar{\boldsymbol{f}}(\boldsymbol{x})]$, the average KL divergence between individual model predictions and the ensemble prediction.

We write:

$$
\begin{aligned}
\mathrm{H}\left[y \,|\, \bar{\boldsymbol{f}}(\boldsymbol{x})\right] &= -\frac{1}{C}\sum_i p(y_i \mid \bar{\boldsymbol{f}}) \log(p(y_i \mid \bar{\boldsymbol{f}})) \\
&= -\frac{1}{C}\sum_i \frac{1}{M}\sum_j p(y_i \mid \boldsymbol{f}_j) \log(p(y_i \mid \bar{\boldsymbol{f}})) \\
&= -\frac{1}{M}\sum_j \frac{1}{C}\sum_i p(y_i \mid \boldsymbol{f}_j) \log(p(y_i \mid \bar{\boldsymbol{f}})) \\
&= -\frac{1}{M}\sum_j \frac{1}{C}\sum_i p(y_i \mid \boldsymbol{f}_j) \left[\log(p(y_i \mid \bar{\boldsymbol{f}})) - \log(p(y_i \mid \boldsymbol{f}_j)) + \log(p(y_i \mid \boldsymbol{f}_j))\right] \\
&= -\frac{1}{M}\sum_j \frac{1}{C}\sum_i p(y_i \mid \boldsymbol{f}_j) \left[\log[\frac{p(y_i \mid \bar{\boldsymbol{f}})}{p(y_i \mid \boldsymbol{f}_j)}] + \log(p(y_i \mid \boldsymbol{f}_j))\right] \\
&= -\frac{1}{M}\sum_j \frac{1}{C}\sum_i p(y_i \mid \boldsymbol{f}_j) \left[\log(\frac{p(y_i \mid \bar{\boldsymbol{f}})}{p(y_i \mid \boldsymbol{f}_j})\right] + -\frac{1}{M}\sum_j \frac{1}{C}\sum_i p(y_i \mid \boldsymbol{f}_j) \log p(y_i \mid \boldsymbol{f}_j)) \\
&= \underset{p(\boldsymbol{f})}{\mathrm{JSD}}\left[y \,|\, \boldsymbol{f}(\boldsymbol{x})\right] + \underset{p(\boldsymbol{f})}{\mathbb{E}}\left[\mathrm{H}\left[y \,|\, \boldsymbol{f}(\boldsymbol{x})\right]\right]
\end{aligned}
$$

C.2 Variance and quadratic uncertainty

As in the main text, we provide a decomposition for a quadratic notion of uncertainty as:

$$
U\left(\bar{\boldsymbol{f}}(\boldsymbol{x})\right) = \underset{p(\boldsymbol{f})}{\mathrm{Var}}\left[\boldsymbol{f}(\boldsymbol{x})\right] + \underset{p(\boldsymbol{f})}{\mathbb{E}}\left[U\left(\boldsymbol{f}(\boldsymbol{x})\right)\right] \tag{8}
$$

where $U(\boldsymbol{f}(\boldsymbol{x}))$ is a quadratic notion of uncertainty:

$$
U\left(\boldsymbol{f}(\boldsymbol{x})\right) \triangleq 1 - \sum_{i=1}^{C}\left[p(y=i \mid \boldsymbol{f}(\boldsymbol{x}))\right]^2.
$$

And variance is defined as:

$$
\underset{p(\boldsymbol{f})}{\mathrm{Var}}\left[\boldsymbol{f}(\boldsymbol{x})\right] = \textstyle\sum_{i=1}^{C} \underset{p(\boldsymbol{f})}{\mathrm{Var}}\left[f^{(i)}(\boldsymbol{x})\right]
$$

Then, the ensemble uncertainty can be decomposed as follows:

$$
\begin{aligned}
U\left(\bar{\boldsymbol{f}}(\boldsymbol{x})\right) &= 1 - \sum_{i=1}^{C}\left[p(y=i \mid \bar{\boldsymbol{f}}(\boldsymbol{x}))\right]^2 \\
&= 1 - \underset{p(\boldsymbol{f})}{\mathbb{E}}[\sum_{i=1}^{C}\left[p(y=i \mid \boldsymbol{f}(\boldsymbol{x}))\right]]^2 + \underset{p(\boldsymbol{f})}{\mathbb{E}}[\sum_{i=1}^{C}\left[p(y=i \mid \boldsymbol{f}(\boldsymbol{x}))\right]]^2 - \sum_{i=1}^{C}\left[p(y=i \mid \bar{\boldsymbol{f}}(\boldsymbol{x}))\right]^2 \\
&= \underset{p(\boldsymbol{f})}{\mathbb{E}}[1 - \sum_{i=1}^{C}\left[p(y=i \mid \boldsymbol{f}(\boldsymbol{x}))\right]]^2 + \sum_{i=1}^{C}\underset{p(\boldsymbol{f})}{\mathbb{E}}[\left[p(y=i \mid \boldsymbol{f}(\boldsymbol{x}))\right]]^2 - \left[\underset{p(\boldsymbol{f})}{\mathbb{E}}[p(y=i \mid \boldsymbol{f}(\boldsymbol{x}))]\right]^2 \\
&= \underset{p(\boldsymbol{f})}{\mathbb{E}}\left[U\left(\boldsymbol{f}(\boldsymbol{x})\right)\right] + \underset{p(\boldsymbol{f})}{\mathrm{Var}}\left[\boldsymbol{f}(\boldsymbol{x})\right]
\end{aligned}
$$

D Brier score Jensen gap

We consider the Brier Score of a single model:

$$
\begin{aligned}
\underset{\boldsymbol{f}}{\mathbb{E}}\left[B_p(\boldsymbol{f}_i)\right] &= \underset{p(\boldsymbol{x},y)}{\mathbb{E}}\,\underset{\boldsymbol{f}}{\mathbb{E}}\left[\|\boldsymbol{f}_i(\boldsymbol{x}) - \mathbf{1}_y\|_2^2\right] \\
&= \underset{p(\boldsymbol{x},y)}{\mathbb{E}}\left[\underset{\boldsymbol{f}}{\mathbb{E}}\left[\|\boldsymbol{f}_i(\boldsymbol{x})\|_2^2\right] + 2\bar{\boldsymbol{f}}_(\boldsymbol{x})^\top \mathbf{1}_y + 1\right]
\end{aligned} \tag{9}
$$

and the Brier score of the ensemble:

$$B_p(\bar{\boldsymbol{f}}) = \mathop{\mathbb{E}}_{p(\boldsymbol{x},y)} \left[\|\bar{\boldsymbol{f}}(\boldsymbol{x}) - \mathbf{1}_y\|_2^2 \right] \tag{10}$$
$$= \mathop{\mathbb{E}}_{p(\boldsymbol{x},y)} \left[\|\bar{\boldsymbol{f}}(\boldsymbol{x})\|_2^2 + 2\bar{\boldsymbol{f}}(\boldsymbol{x})^\top \mathbf{1}_y + 1 \right]$$

Note that Eq. (9) and Eq. (10) only differ by a single term:

$$\begin{aligned} B_p(\bar{\boldsymbol{f}}) &= \mathop{\mathbb{E}}_{\boldsymbol{f}} [B_p(\boldsymbol{f})] + \mathop{\mathbb{E}}_{p(\boldsymbol{x})} \left[\|\bar{\boldsymbol{f}}(\boldsymbol{x})\|_2^2 \right] \\ &\quad - \mathop{\mathbb{E}}_{p(\boldsymbol{x})} \mathop{\mathbb{E}}_{\boldsymbol{f}} \left[\|\boldsymbol{f}(\boldsymbol{x})\|_2^2 \right] \\ &= \mathop{\mathbb{E}}_{\boldsymbol{f}} [B_p(\boldsymbol{f})] - \mathop{\mathbb{E}}_{p(\boldsymbol{x})} [\| \mathop{\mathbb{E}}_{\boldsymbol{f}} \boldsymbol{f}(\boldsymbol{x})\|_2^2 - \|\bar{\boldsymbol{f}}(\boldsymbol{x})\|_2^2] \\ &= \mathop{\mathbb{E}}_{\boldsymbol{f}} [B_p(\boldsymbol{f})] - \mathop{\mathbb{E}}_{p(\boldsymbol{x})} [\mathop{\mathbb{E}}_{\boldsymbol{f}} [\|\boldsymbol{f}(\boldsymbol{x})\|_2^2] - \| \mathop{\mathbb{E}}_{\boldsymbol{f}} [\boldsymbol{f}(\boldsymbol{x})]\|_2^2] \\ &= \mathop{\mathbb{E}}_{\boldsymbol{f}} [B_p(\boldsymbol{f})] - \mathop{\mathbb{E}}_{p(\boldsymbol{x})} [\mathop{\mathrm{Var}}_{p(\boldsymbol{f})} [\boldsymbol{f}(\boldsymbol{x})]] \end{aligned}$$

Where $\mathrm{Var}_{p(\boldsymbol{f})} [\boldsymbol{f}(\boldsymbol{x})]$ is:

$$\mathop{\mathrm{Var}}_{p(\boldsymbol{f})} [\boldsymbol{f}(\boldsymbol{x})] = \textstyle\sum_{j=1}^{C} \mathop{\mathrm{Var}}_{p(\boldsymbol{f})} \left[f^{(j)}(\boldsymbol{x}) \right]$$

.

We note that this relation holds at the level of individual data points as well.

E Expected behavior of Bayesian model average on InD/OOD uncertainty quantification

As a motivating example, we consider uncertainty quantification on InD and OOD data using a Bayesian model average, and relate our findings back to the implications presented in Sec. 4.5. An ideal Bayesian model average should express higher posterior variance on OOD data than InD data, even after controlling for other sources of uncertainty. To demonstrate this desired behavior in practice, we consider Gaussian processes, a class of models well regarded for its uncertainty quantification capabilities [65]. The Gaussian process model $f(\cdot)$ is defined by the following generative process:

$$\begin{aligned} p(f(\cdot)) &= \mathcal{GP}, \\ p(y \mid f(x)) &= \mathcal{N}(0, \sigma^2(x)) \end{aligned} \tag{11}$$

where $\sigma^2(x)$ is a heteroskedastic noise function defined as $\sigma^2(x) = \sin^2(x) + 0.01$. After conditioning on training data $\mathcal{D}$, the BMA at a test point x is given by:

$$p(y \mid x, \mathcal{D}) = \mathcal{N}(\mu_{f|\mathcal{D}}(x), \mathrm{Var}_{f|\mathcal{D}}(x) + \sigma^2(x)), \tag{12}$$

where $\mu_{|\mathcal{D}}(\cdot)$ and $\mathrm{Var}_{f|\mathcal{D}}(\cdot)$ are the posterior predictive GP mean and variance, respectively, which can both be computed in closed form. (See [65] for closed-form expressions for these two functions). Crucially, the predictive variance in Eq. (12) is a uncertainty estimate that decomposes into epistemic and aleatoric components: the **posterior variance** term ($\mathrm{Var}_{f|\mathcal{D}}(\cdot)$) and the **likelihood variance** term ($\sigma^2(x)$), respectively.

In Fig. 5 (left), we generate a one-dimensional dataset by drawing 25 random data points over $x \in [0, 5]$ using the generative process defined in Eq. (11).[2] After fitting a GP model to these data, we compute the predictive posterior over the range $x \in [-5, 5]$. The points in $[0, 5]$ represent InD data—as they share the same domain as the training data—while the points in $[-5, 0]$ (orange) represent OOD data. In Fig. 5 (right), we observe that OOD predictions have much higher expected posterior variance, even after conditioning on a prediction's likelihood uncertainty. Note that this is in stark contrast to the analogous deep ensemble results in Sec. 4, where there is little to no conditional difference between OOD and InD predictions.

[2] In all experiments, the prior GP model has zero mean and a RBF covariance function with a lengthscale of 1.

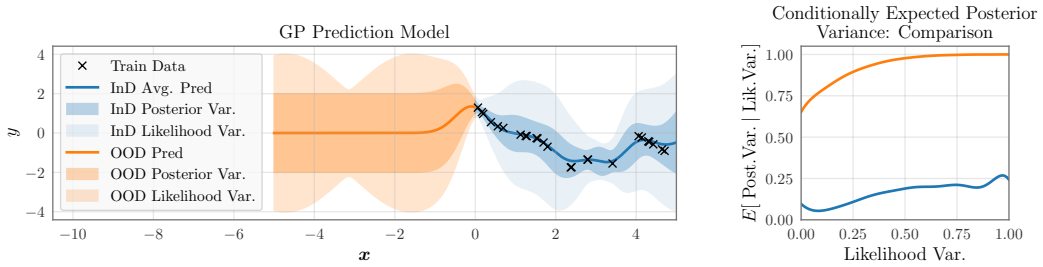

Figure 5: Example of a model where OOD predictions have higher posterior variance, even after controlling for other sources of uncertainty. **Left:** The predictive uncertainty expressed by a Gaussian process model on a toy regression dataset. OOD data (orange) express higher posterior variance than InD data (blue). **Right:** The expected posterior variance) conditioned on a prediction's likelihood variance is also significantly larger for OOD data.

F Quantifying conditional diversity

In this section, we provide additional experimental details for the results in Fig. 1, and extend to other datasets and measures of ensemble diversity. We also introduce quantifications and signficance tests to validate the stability of our conclusions across many combinations of OOD dataset and model.

F.1 Marginal distribution of average single model uncertainty

We end Sec. 4.2 with the surprising conclusion that any changes to ensemble UQ between InD and OOD data must come from changes in the distribution of average single model uncertainty, $p(\mathbb{E}(U(f(x))$. Here we confirm empirically that this distribution does shift towards higher uncertainty on OOD data, for the same models that we present in Sec. 4.2. This shift drives any changes in ensemble diversity that we observe in practice.

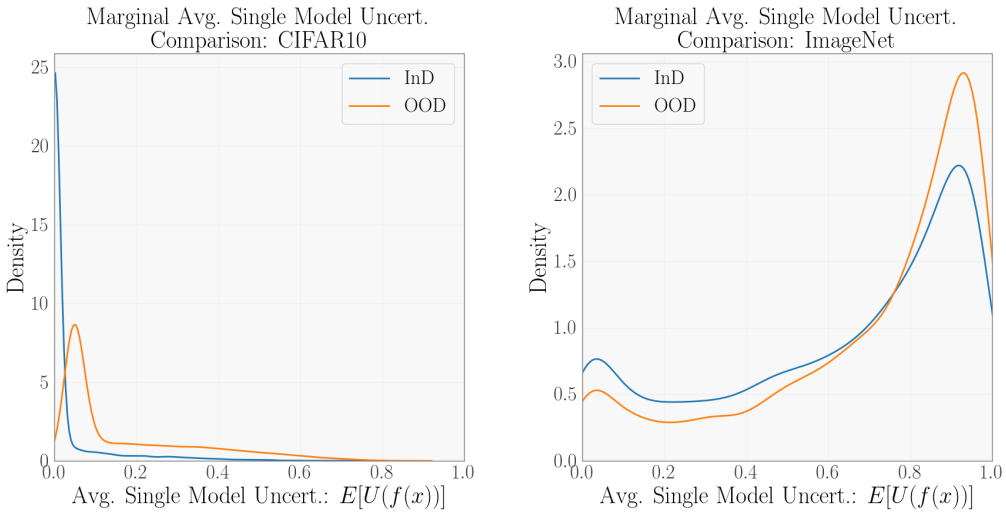

Figure 6: Distributions of average single model uncertainty for the WideResNet 28-10 ensembles trained on CIFAR10 (left) and the AlexNet ensembles (right), as in Fig. 1. InD and OOD test datasets are CIFAR10 and CINIC10 for the left panel, and ImageNet and ImageNet V2 for the right.

F.2 Generating conditional distributions and conditional expectations

In order to depict conditional variance distributions, we fit kernel density estimates to the joint distribution of ensemble diversity and average single model uncertainty for all evaluation datasets. We generated KDEs with the bandwidth suggested by Scott's Rule, and approximate conditional distributions by dividing each column of our KDE grid by the average value.

To validate comparisons between conditional distributions more precisely, we estimate the conditional expectation $\mathbb{E}[\text{Diversity} \mid \text{Avg}]$ by fitting a Kernel Ridge Regression model to these data, giving the best fit curve to predict values of ensemble diversity from a given value of average single model uncertainty. We used a Gaussian kernel, with bandwidth identical to what was used to generate KDE plots.

Strictly to ease visualization, we generated conditional expectation estimates for CINIC10 with a randomly subsampled set of 10000 points when fitting Kernel Ridge Regression. We account for any potential bias this may introduce in our statistical quantifications below.

F.3 Visualizations for other datasets and metrics

Figure 7 first shows the variance analysis that we conducted extended to CIFAR10/CIFAR10.1, estimated with an ensemble of 5 VGG 11 networks. In the rows below, we show all analogous conclusions for Jensen Shannon Divergence as a measure of ensemble diversity, instead of variance for the same models (ensembles of $M = 4$ VGG-11, WideResNet28-10, and AlexNet models for CIFAR10.1, CINIC10 and ImageNet V2 respectively). Across all datasets, we observe that the same trends hold as reported in Fig. 1. Namely, ensemble diversity is higher on OOD data than InD data, but that the corresponding conditional distributions are not distinguishable.

F.4 Large scale quantification and statistical tests

In order to scale these analyses further, we devised a test statistic to directly compare the conditional expected diversity measures of InD and OOD data. Given conditional expectations for InD and OOD data, consider the following statistic:

$$d(InD, OOD) = \int d\text{Avg} \frac{\mathbb{E}_{OOD}[\text{Diversity} \mid \text{Avg}] - \mathbb{E}_{InD}[\text{Diversity} \mid \text{Avg}]}{\mathbb{E}_{InD}[\text{Diversity} \mid \text{Avg}]}$$

Intuitively, this statistic measures the percentage change in area under the conditional expectation curve when we consider an OOD conditional expectation instead of a corresponding InD conditional expectation.

We approximated this percentage increase in expected conditional diversity as sum of pointwise differences between InD and OOD, divided by the sum of the InD curve, and report results for all model and dataset pairs that we tested in Fig. 8,Fig. 9. Altogether, we see that in most cases, the percentage increases in area under the OOD curve are very small (for reference, the main text examples demonstrate changes on the order of $\sim 1\%$.) Although there are few sporadic cases where certain datasets demonstrate sizeable increases in our statistic on OOD data (consider variance for DenseNet 169 on CIFAR10-C Gaussian Noise, Severity Level 5), we note that these trends are inconsistent across individual models and datasets, limiting practical use of differences in OOD estimation. Furthermore, we note that our results on natural corruptions (leftmost two columns) are far more consistent than our results on synthetic corruptions (all others). In line with previous work [72], we prioritize results on natural corruptions in reporting our results.

Next, we performed Monte Carlo permutation tests to quantify the significance of the statistics that we observed:

- For each model and dataset upon which we computed a statistic, we first aggregated all datapoints from in and out of distribution model evaluations, and randomly permuted the order of these samples, generating a surrogate sample.
- We then refit Kernel Ridge Regression to the surrogate sample, and calculated the d statistic that resulted.
- We calculated if the computed d statistic was greater than or less than what we observed on our original sample.

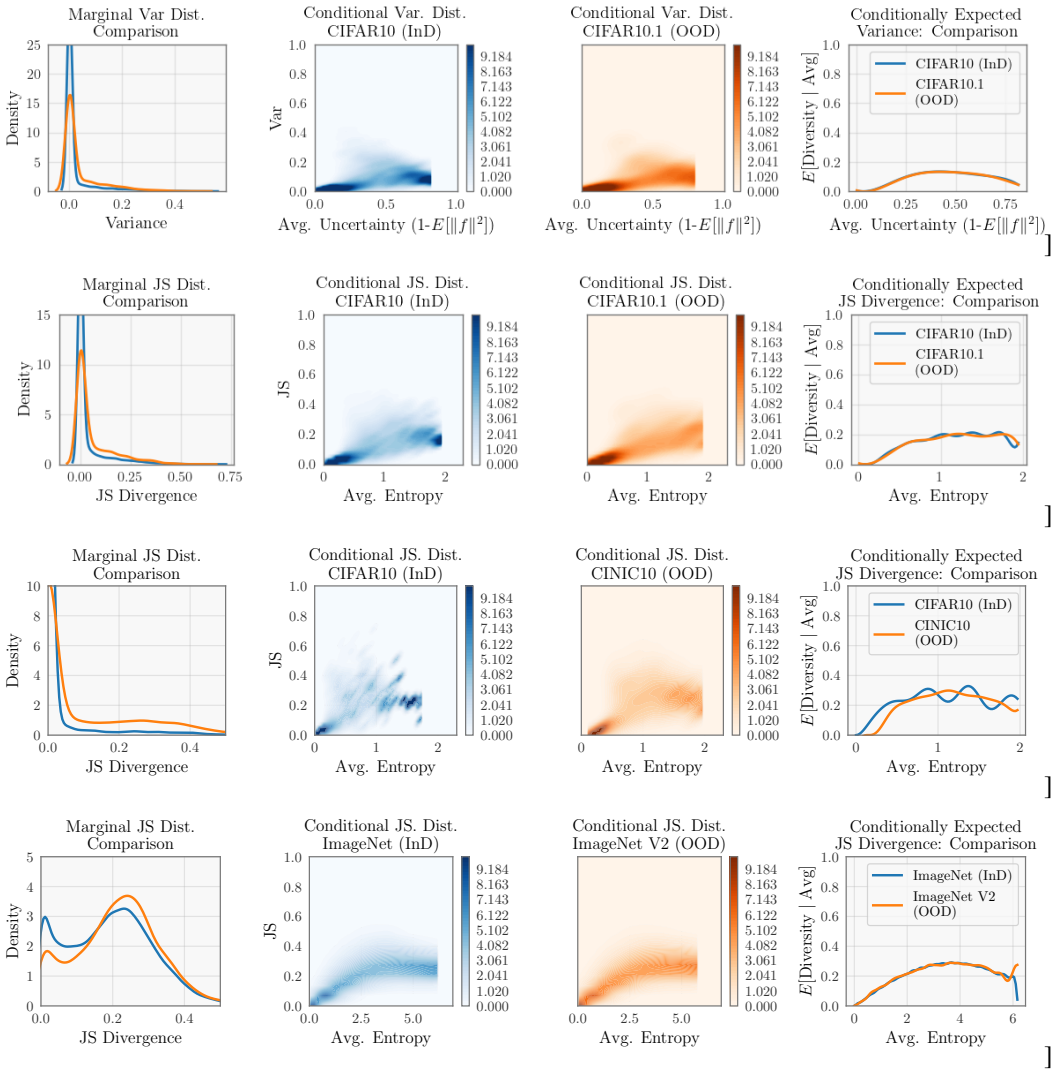

Figure 7: The top panels illustrates the InD vs OOD Variance for Cifar 10 vs Cifar10.1 with an ensemble of 4 VGG-11 networks. The bottom 3 panels illustrate the JS divergence on InD (Blue) and OOD (orange) data for CIFAR10 vs CIFAR10.1 (VGG-11), CIFAR10 vs CINIC10 (WideResNet-28-10) and ImageNet vs ImageNetV2 (AlexNet). Conventions and conclusions as in Figure 1.

- We repeated this process for a total of 100 surrogate samples.

From this process, we can treat the proportion of surrogate samples that exceeded the value of our true test statistic as a p value for the null hypothesis that the d statistic we calculated measures a significant difference between our two original samples (and in particular, that the conditional expectation of ensemble diversity on OOD data is significantly greater than that of ensemble diversity in InD data.)

In order to compute kernel ridge regression efficiently, we used GPytorch [23] with kernel partitioning to refit models many times on a GPU. This process allowed us to compute statistics on the entire CINIC10 evaluation set, alleviating all possibilities for error in visualization due to subsampling.

In Fig. 10 and Fig. 11, we report the estimated p values from this process. Our main goal is to communicate that in many cases, we found that the differences between conditional expectations for in and out of distribution data were almost certainly not significant, regardless of their absolute magnitude.

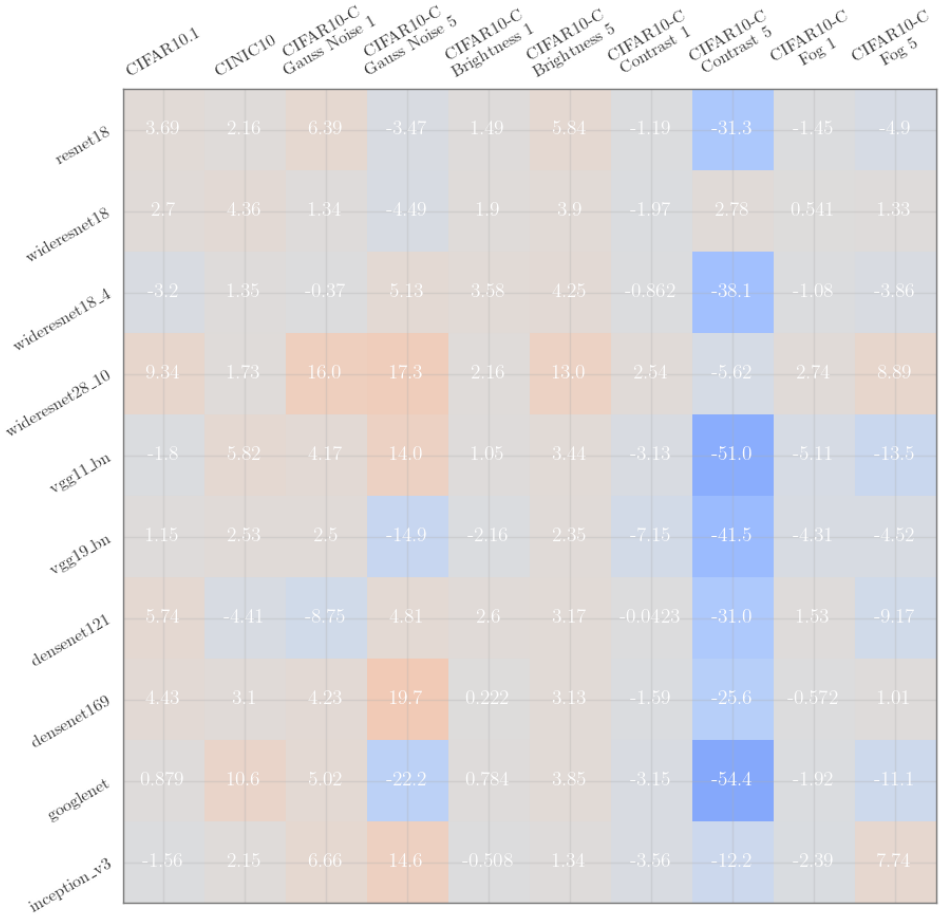

Figure 8: Percent Increase (OOD over InD) for Variance Decomposition

Finally, we show percentage increases for Imagenet on analogous $M = 5$ ensembles of AlexNet, ResNet 50, and ResNet 101 models Fig. 10, Fig. 11- on ImageNet V2, we once again fail to see any considerable increase on the conditional distributions of OOD data relative to InD data, regardless of metric.

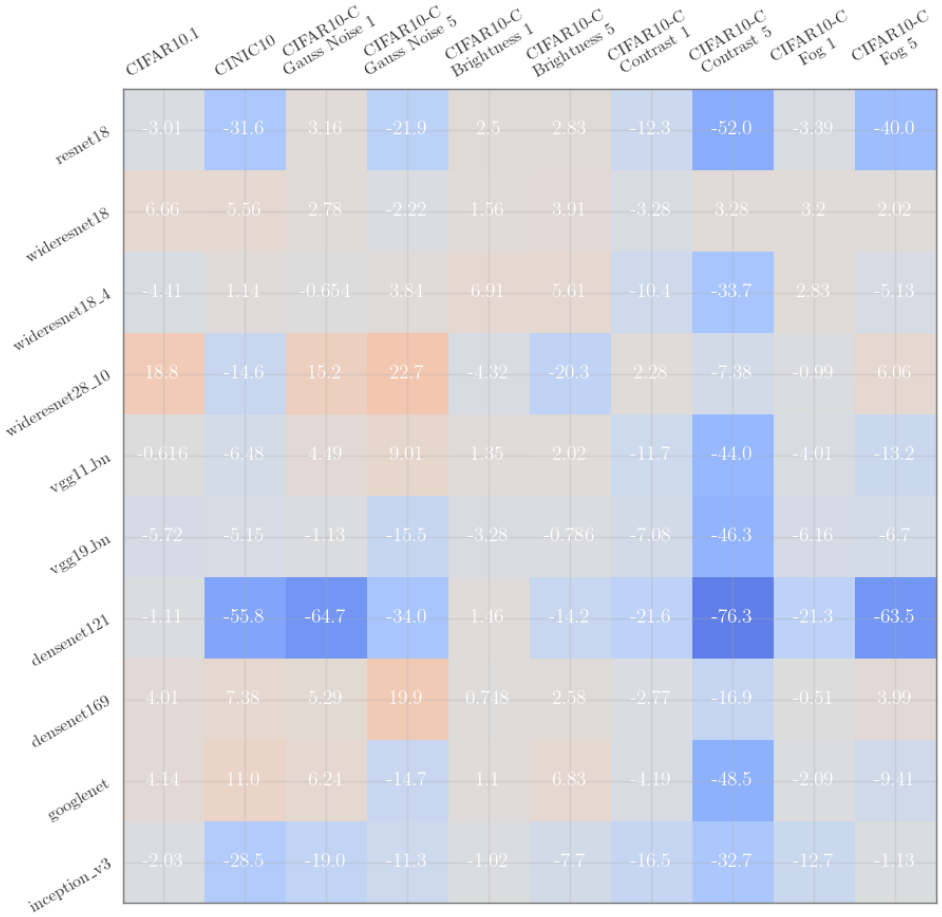

Figure 9: Percent Increase (OOD over InD) Jensen Shannon Decomposition

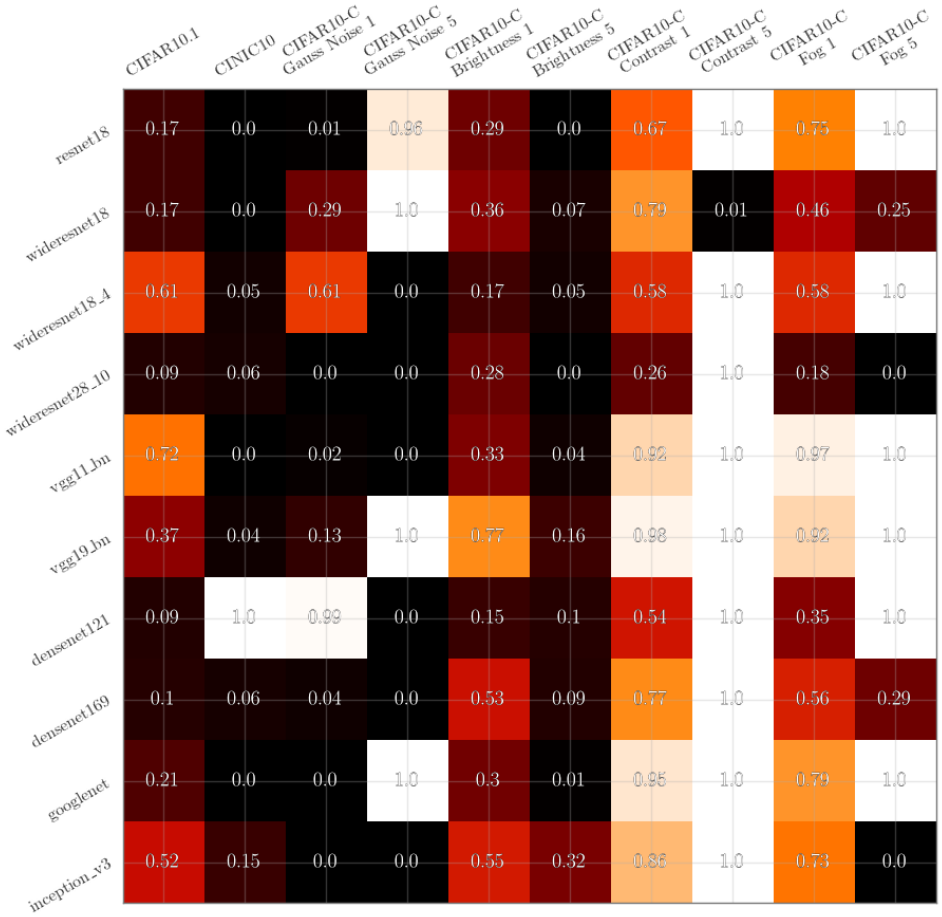

Figure 10: P values of (OOD over InD) difference for Variance Decomposition

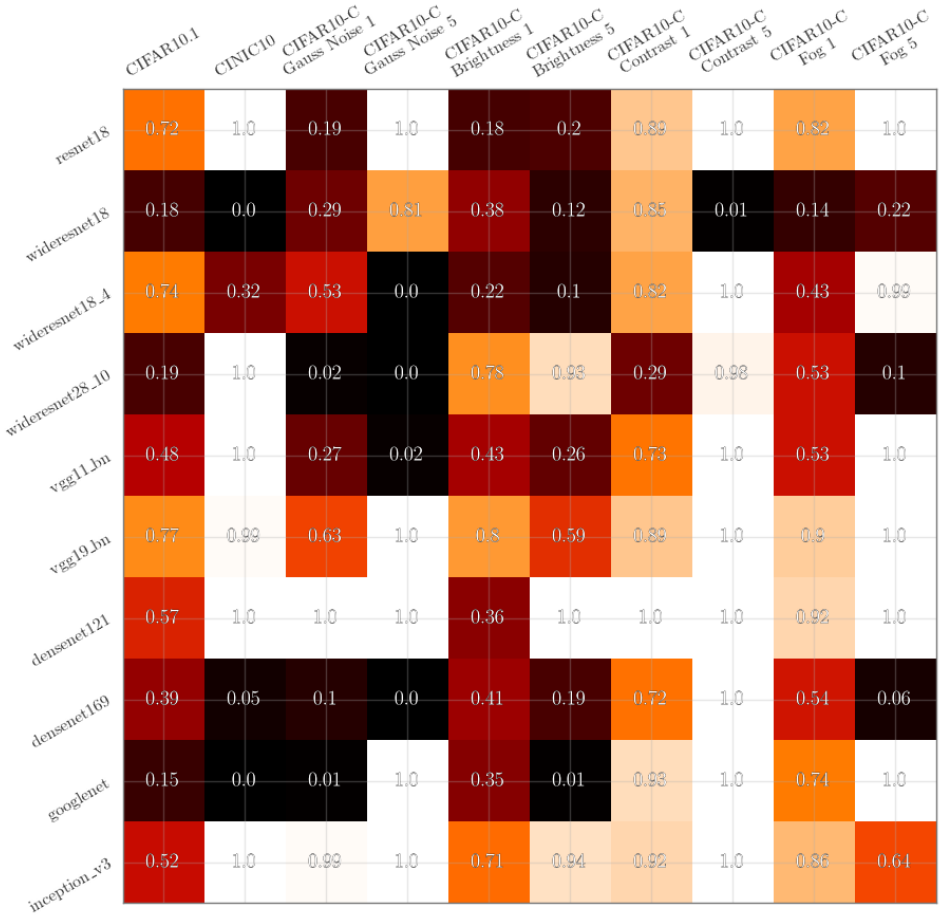

Figure 11: P values of (OOD over InD) difference for Jensen Shannon Decomposition

We can replicate the finding that differences between in and out of distribution test sets are quite small in the ImageNet dataset as well:

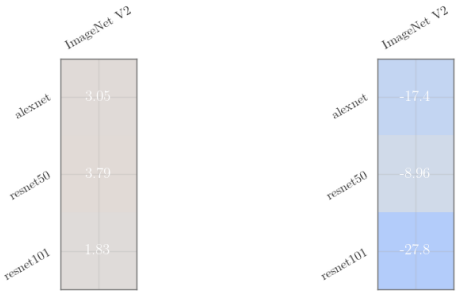

Figure 12: Percent Increase (OOD over InD) Variance Decomposition (left) and JS Divergence (right)

G Details of models for robustness experiments

We followed many of the same experimental procedures as [52] in order to generate ensembles for our experiments. We denote four main groups of models below:

G.1 CIFAR10 models trained from scratch

We trained 10 different classes of models on CIFAR10, noted below. We used implementations from https://github.com/huyvnphan/PyTorch_CIFAR10 in order to train convolutional models adapted for CIFAR10 data sizes, with default hyperparameters, and manually extended existing implementations in this repo to create a WideResNet 18 with width 4.

- ResNet 18 [33]
- WideResNet 18-2, 18-4, 28-10 [78]
- GoogleNet, Inception v3 [71]
- VGG with 11 and 19 layers [69]
- DenseNet 121 and 169 [37]

We trained five independent instances of each of these architectures with random seeds for 100 epochs each (see code repo defaults for other hyper parameters.)

G.2 CIFAR10 pretrained ensembles

We use the models trained by Miller et al. [52], and we thank the authors for graciously sharing these results with us.

G.3 ImageNet models trained from scratch

We additionally trained two sets of ensembles from scratch on the ImageNet dataset. In particular, we trained 5 model ensembles of AlexNet and ResNet 101 models using implementations available at https://pytorch.org/vision/stable/models.html for 90 epochs each.

G.4 Imagenet pretrained models

We use 5 of the ResNet50 models trained by [4] and the standard 78 trained models provided by Taori et al. [72].

H Additional generalization trend results

In this section, we report test statistics for the results we show in Fig. 4, and we extend the results from Fig. 4 to additional OOD datasets, namely CIFAR10.1 and ImageNet-C [34], illustrating generalization trends for ensembles and individual models for various distortions at different intensity levels. The results in this section show that for high intensity distortions, single models can break away from a well defined linear trend, as reported in [52]. However, even at the highest distortion levels, the generalization performance for ensembles and individual models heavily overlap, suggesting the lack of effective robustness demonstrated by deep ensembles is not dependent upon the same phenomena that generate strong trends in single models to begin with.

H.1 Test statistics for generalization performance trends

In each table we report the regression coefficient (Coefficient), the standard error (Std. error) t-statistic, p-value and R^2 to reject the null hypothesis that there is no relation between InD and OOD performance for the different metrics considered (left column). The last column indicates the number of models (markers) for each model class depicted in Fig 4.

Note that we do not apply logit scaling to our axes as in [72], which was found to increase the fit of linear trend lines. Furthermore, we do not consider non-linear parametrizations of NLL, which could

potentially improve the quantification of overlap between single models and ensembles. We consider such parameterizations to be beyond the scope of this work.

Table 1: R^2 **for InD vs OOD generalization trend fits for different metrics**: CIFAR10 vs CINIC10 in Fig. 4a.

Metric	Type	Coefficient	Std. error	t-statistic	p-value	R^2	Number of models
0-1 Error	All	0.038	0.002	18.981	0.0	0.853	434
	Single Model	0.029	0.006	5.038	0.0	0.883	54
	Ensemble	0.039	0.002	18.349	0.0	0.848	380
NLL	All	0.116	0.006	18.285	0.0	0.894	434
	Single Model	0.120	0.022	5.511	0.0	0.864	54
	Ensemble	0.116	0.007	17.559	0.0	0.896	380
Brier	All	0.051	0.003	17.415	0.0	0.876	434
	Single Model	0.042	0.009	4.754	0.0	0.890	54
	Ensemble	0.052	0.003	16.754	0.0	0.873	380
rESCE	All	0.009	0.002	4.712	0.0	0.791	434
	Single Model	0.026	0.007	3.755	0.0	0.632	54
	Ensemble	0.007	0.002	3.860	0.0	0.801	380

Table 2: R^2 **for InD vs OOD generalization trend fits for different metrics**: ImageNet vs ImageNetV2 in Fig. 4b.

Metric	Type	Coefficient	Std. error	t-statistic	p-value	R^2	Number of models
0-1 Error	All	0.102	0.001	89.935	0.0	0.995	367
	Single Model	0.105	0.002	43.326	0.0	0.994	93
	Ensemble	0.101	0.001	78.643	0.0	0.995	274
NLL	All	0.432	0.008	54.749	0.0	0.989	367
	Single Model	0.443	0.018	24.091	0.0	0.984	93
	Ensemble	0.428	0.009	49.622	0.0	0.991	274
Brier	All	0.156	0.002	77.827	0.0	0.989	367
	Single Model	0.159	0.005	34.540	0.0	0.985	93
	Ensemble	0.156	0.002	69.984	0.0	0.991	274
rESCE	All	0.060	0.003	19.723	0.0	0.111	367
	Single Model	0.067	0.006	10.871	0.0	0.090	93
	Ensemble	0.058	0.004	16.342	0.0	0.113	274

Table 3: R^2 **for InD vs OOD generalization trend fits for different metrics**: CIFAR10 vs CIFAR10.1 in Fig. 13

Metric	Type	Coefficient	Std. error	t-statistic	p-value	R^2	Number of models
0-1 Error	All	0.038	0.002	18.981	0.0	0.853	434
	Single Model	0.029	0.006	5.038	0.0	0.883	54
	Ensemble	0.039	0.002	18.349	0.0	0.848	380
NLL	All	0.116	0.006	18.285	0.0	0.894	434
	Single Model	0.120	0.022	5.511	0.0	0.864	54
	Ensemble	0.116	0.007	17.559	0.0	0.896	380
Brier	All	0.051	0.003	17.415	0.0	0.876	434
	Single Model	0.042	0.009	4.754	0.0	0.890	54
	Ensemble	0.052	0.003	16.754	0.0	0.873	380
rESCE	All	0.009	0.002	4.712	0.0	0.791	434
	Single Model	0.026	0.007	3.755	0.0	0.632	54
	Ensemble	0.007	0.002	3.860	0.0	0.801	380

H.2 Evaluation on other datasets

In this section we follow the same conventions as in Fig. 4 to analyze the generalization performance for two other OOD datasets for CIFAR10 and ImageNet, namely CIFAR10.1 and ImageNetC [34]. For ImageNetC we focus on our distortions from this dataset; namely brightness, contrast, fog and gaussian noise for three different degrees of corruption.

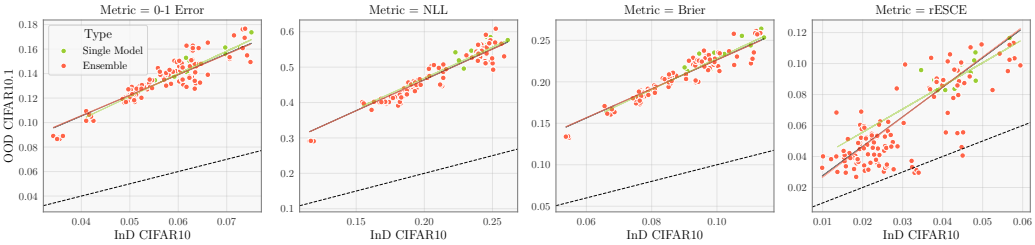

Figure 13: Generalization Trends for CIFAR10 vs CIFAR10.1.
Conventions and conclusions as in Fig. 4.

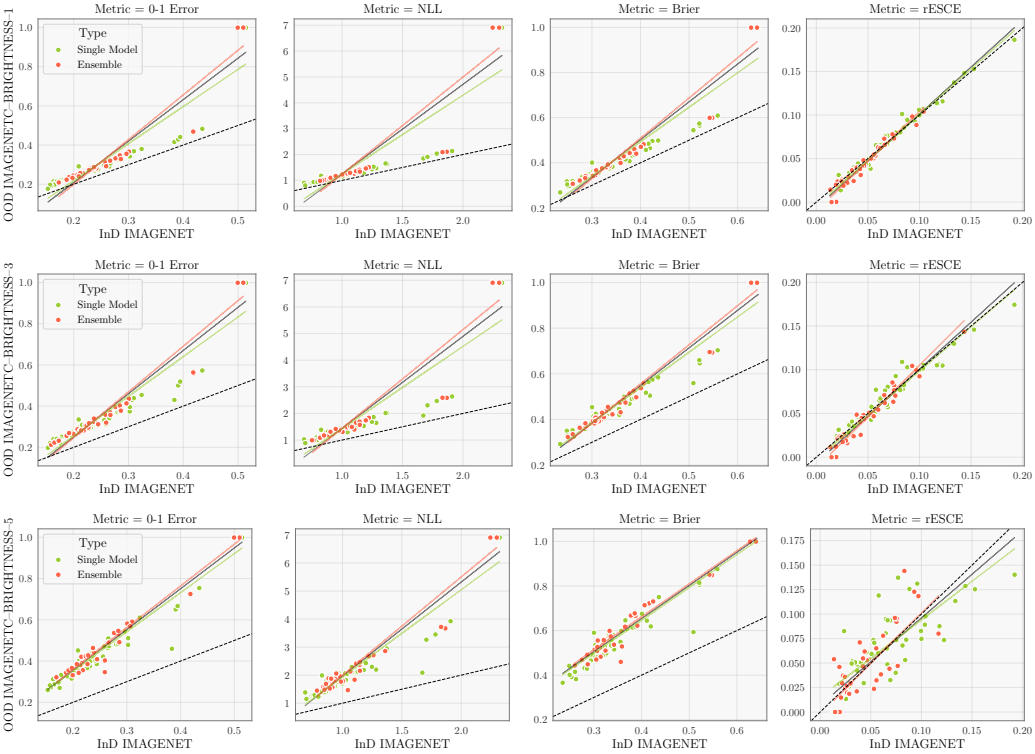

Figure 14: Generalization Trends for ImageNet vs ImageNet-C Brightness-1, 3 and 5.
Conventions and conclusions as in Fig. 4.

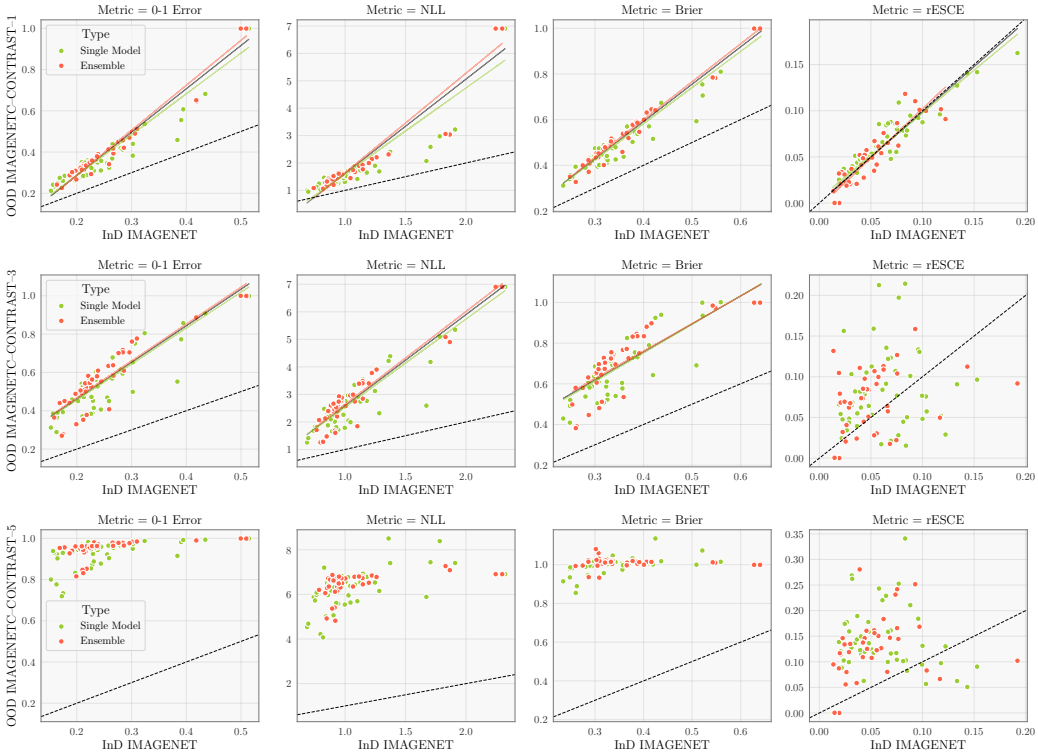

Figure 15: Generalization Trends for ImageNet vs ImageNet-C Contrast-1, 3 and 5. Conventions and conclusions as in Fig. 4.

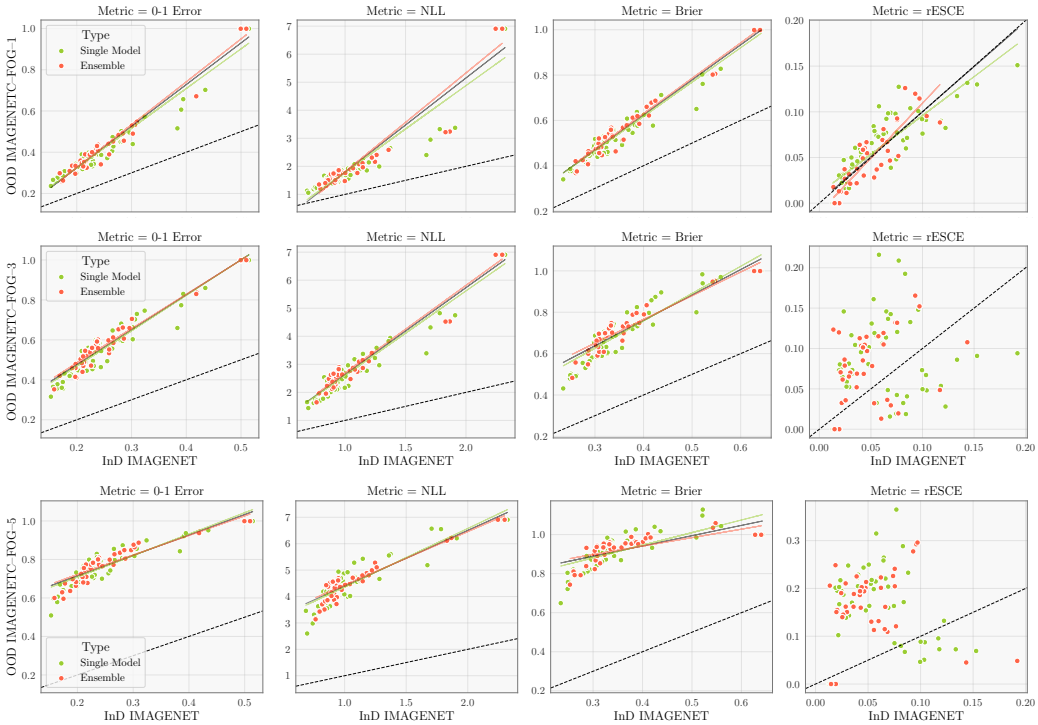

Figure 16: Generalization Trends for ImageNet vs ImageNet-C Fog-1,3, and 5. Conventions and conclusions as in Fig. 4.

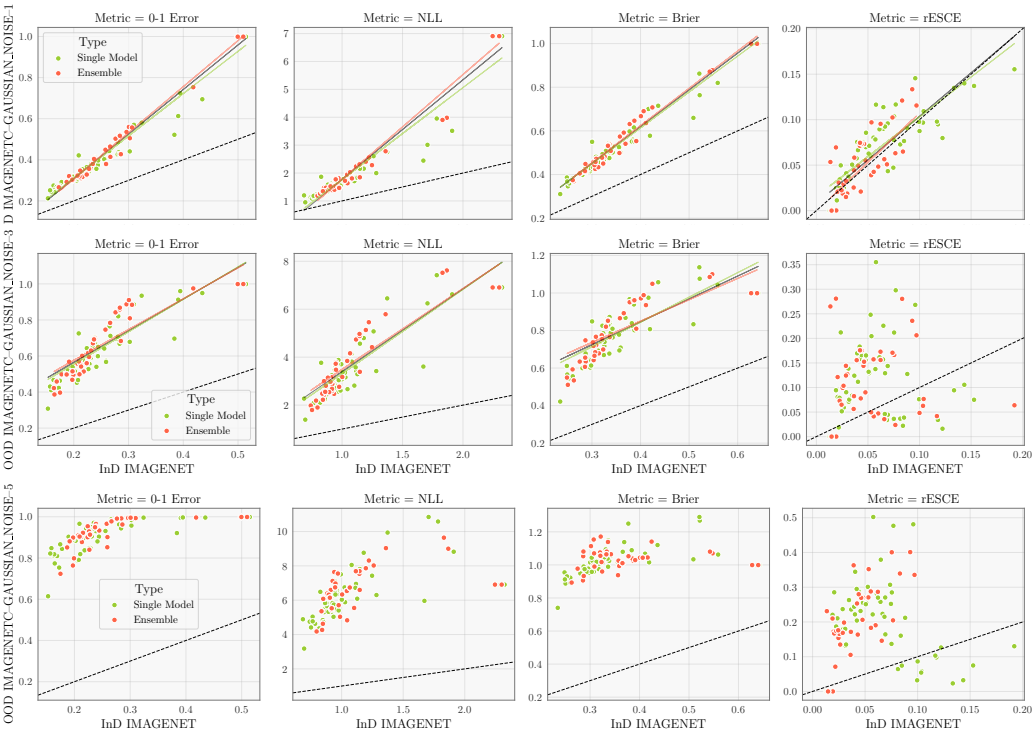

Figure 17: Generalization Trends for ImageNet vs ImageNet-C Gaussian Noise-1,3, and 5. Conventions and conclusions as in Fig. 4.

H.3 Calibration metrics

The calibration error is a frequentist idea to measure the quality of uncertainty given by a model. The calibration error is given by,

$$\text{Calibration Error} = |Confidence - Accuracy|$$

To compare the calibration error across multiple models, practitioners have resorted to the expected calibration error (ECE) [58], which approximates the calibration error by binning the predictive probabilities and taking a weighted average of the calibration errors across bins. ECE provides a scalar summary statistic of the quality of uncertainty [26]. We employ both the ECE and, a smooth approximation, the root of the Expected Squared Calibration Error (rESCE) to compare the quality of uncertainty gained by ensembling over individual models. The rESCE is defined as,

$$\text{rESCE} = \sqrt{\sum \frac{|B_m|}{n} [acc(B_m) - conf(B_m)]^2}$$

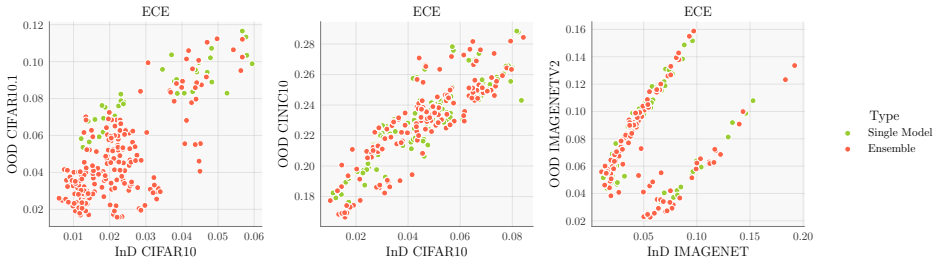

Figure 18: Generalization trends for the Expected Calibration Error (ECE) for different datasets. Conventions and conclusions similar to the rESCE (right column) in Fig. 4.

Fig. 4 provides the rESCE for in distribution vs out of distribution for CIFAR10 vs CIFAR10.1 and Imagenet vs ImagenetV2. (See Appx. H.2 for additional datasets). In Fig. 18, we evaluate the generalization performance in terms of the ECE, following the conventions in Fig. 4. From Fig. 18 we see that there is no clear trend for InD versus OOD generalization across different datasets. Furthermore, we find that—for CIFAR10/CIFAR10.1—ensembles are able to achieve some amount of effective robustness with respect to the ECE metric. However, for the other two dataset pairs, we find that the ECE performance of ensembles heavily overlaps with the ECE performance of single models for most models.

H.4 Comparison between homogeneous, heterogeneous and implicit deep ensembles

In this section we split the data from the ensemble model class in Fig. 4 into two sub classes: an ensemble class which now contains only the homogeneous ensembles, and the heterogeneous class; to explore if the ensemble model classes provide different generalization trends. We find that this is not the case for several in distribution and out of distribution pairs illustrated in Fig. 19.

We were additionally interested in evaluating whether implicit deep ensembles, models which aim to bridge the gap between individual networks and deep ensembles such as MC Dropout [21], Batch Ensemble [75], and MIMO [30], also follow the same observed trends. We include the performance of implicit ensembles, including MC Dropout and MIMO, in Fig. 19. The implicit deep ensemble models for ImageNet were constructed from a Resnet50 architecture, which was selected given its ubiquitous deployment, and availability in the open source Uncertainty Baselines library [57]. The number of implicit models considered for ImageNet is 12 (2 MC dropout models, and 10 for MIMO models). The implicit deep ensemble models for CIFAR10 were constructed from a WideResnet-28 architecture. In total, we considered 6 implicit ensemble models (3 MC dropout models and 3 MIMO models). Overall the results illustrated in Fig. 19 show that implicit deep ensembles also fall on the line, along with individual models, heterogeneous, and homogeneous ensembles, and do not constitute an effectively robust model class.

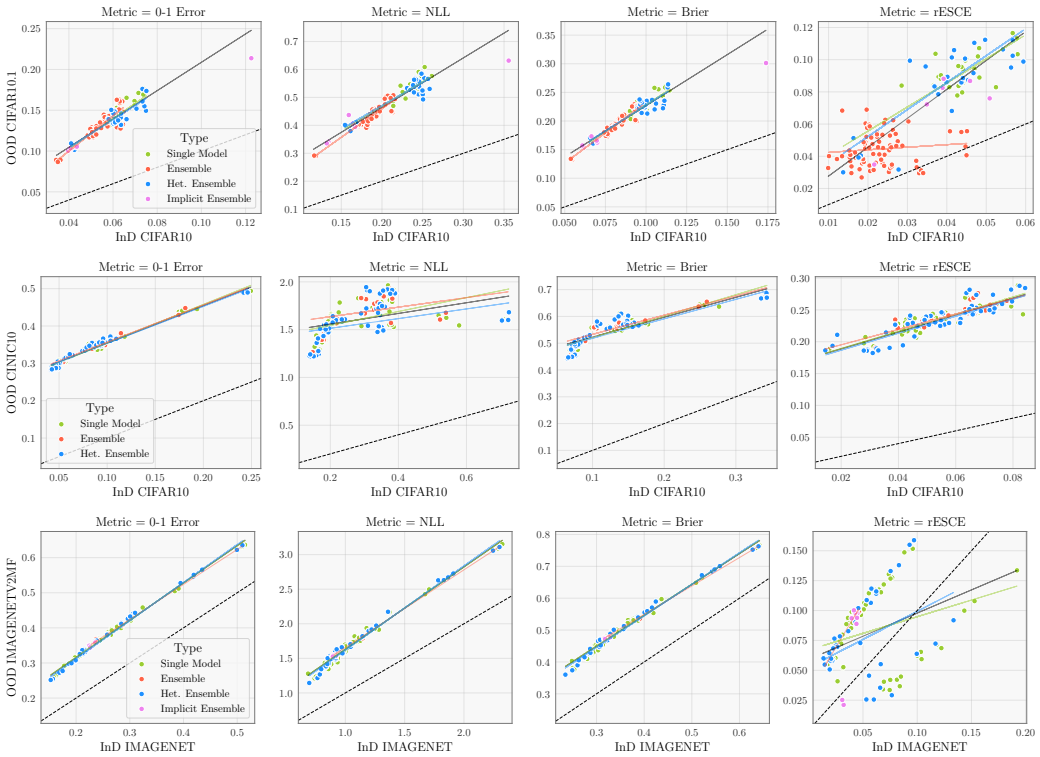

Figure 19: Homogeneous and heterogeneous deep ensembles in Fig. 4 follow similar generalization trends. Conventions and conclusions as in Fig. 4, including Implicit Deep ensemble models.

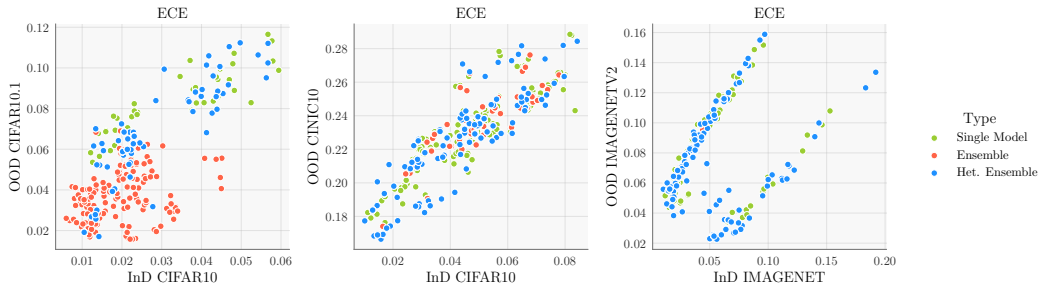

Figure 20: Expected Calibration Error (ECE) for homogeneous and heterogeneous ensembles follow similar trends as rESCE in Fig. 4.

I Quantifying improvement similarity between single models and ensembles

In order to validate the statistical significance of the correlations that we observe between the improvements made by ensembles and single models, we considered as baselines the distribution of improvements we would expect from comparing *within* each model type- i.e. improvement correlations comparing the improvement of two performance matched single models, or two performance matched ensembles. Although Pearson's R provides a good visual aid to interpret the trends visually, we wanted to be more agnostic when validating the trends that we see. We directly compared the improvement correlations that we see between ensemble/larger model pairs with improvement correlations that resulted when we substituted one member of these pairs with another, similarly performing model from the opposite model type (i.e., replace the ensemble with a control single model, or vice versa). We compared the resulting pair of improvement correlations with a kernel two sample test [25]. We calculated the unbiased test statistic MMD_u^2 from this paper for each combination of improvement correlations, and determined an appropriate threshold for these statistics based upon which we would reject the null hypothesis (Corollary 11 in [25]).

For each entry in the tables shown here, we consider the performance of four different kinds of predictions: an ensemble, an average single model with similar performance, a "control" set of average single models or ensembles (again with similar performance) and finally a base model, against which we are comparing the performance of all other models. Each entry compares one of the improvement correlations shown in Fig. 2 (given by the row) against the "control" improvement correlation listed in the column. In all comparisons, across InD and OOD data, CINIC10 and CIFAR10.1, and using NLL or Brier Score as metrics, we failed to reject the null hypothesis that the distributions we compared were significantly different. Each table shows the statistic value that we computed from any given pair of improvement correlations, along with the threshold statistic value we would have to exceed to reject the null hypothesis in parentheses. We list details of each comparison with each table.

Finally, at the end of this section we list the accuracies of the models that we compare, ensuring that the overall performance of these models does not differ too significantly, regardless of the metric under consideration.

Table 4: MMD_u^2 **to compare performance gains: CINIC10 NLL.** Base network: VGG-11. Average Model: WideResNet-18-4. Control Single Model: WideResNet-18-2. Control Ensemble: GoogleNet

		Δ Single/Δ Ctrl. Single	Δ Ensemble/Δ Ctrl. Ensemble
Δ Ensemble/Δ Single	CIFAR10 (InD)	$2.2\text{e}{-3}(0.069)$	$2.1\text{e}{-2}(0.069)$
	CINIC10 (OOD)	$3.3\text{e}{-3}(0.031)$	$2.0\text{e}{-3}(0.031)$

Table 5: MMD_u^2 **to compare performance gains: CINIC10 Brier Score**. Base network: VGG-11. Average Model: WideResNet-18-4. Control Single Model: WideResNet-18-2. Control Ensemble: GoogleNet

		Δ Single/Δ Ctrl. Single	Δ Ensemble/Δ Ctrl. Ensemble
Δ Ensemble/Δ Single	CIFAR10 (InD)	$2.7\text{e}{-4}(0.069)$	$1.3\text{e}{-2}(0.069)$
	CINIC10 (OOD)	$4.2\text{e}{-3}(0.031)$	$4.8\text{e}{-3}(0.031)$

Table 6: MMD_u^2 **to compare performance gains: CIFAR10.1 NLL**. Base network: ResNet 18. Average Model: WideResNet 18-4. Control Single Model: WideResNet 18. Control Ensemble: VGG 11

		Δ Single/Δ Ctrl. Single	Δ Ensemble/Δ Ctrl. Ensemble
Δ Ensemble/Δ Single	CIFAR10 (InD)	$2.4\text{e}{-2}(0.069)$	$5.1\text{e}{-3}(0.069)$
	CIFAR10.1 (OOD)	$8.1\text{e}{-3}(0.15)$	$2.0\text{e}{-3}(0.15)$

Table 7: MMD_u^2 **to compare performance gains: CIFAR10.1 Brier Score**. Base network: ResNet 18. Average Model: WideResNet 18-4. Control Single Model: WideResNet 18. Control Ensemble: VGG 11

		Δ Single/Δ Ctrl. Single	Δ Ensemble/Δ Ctrl. Ensemble
Δ Ensemble/Δ Single	CIFAR10 (InD)	$2.1\text{e}{-2}(0.069)$	$2.5\text{e}{-3}(0.069)$
	CIFAR10.1 (OOD)	$8.1\text{e}{-3}(0.15)$	$8.7\text{e}{-4}(0.15)$

Table 8: **Corresponding accuracies for models compared on CIFAR10/CINIC10.** Architectures of models left to right: VGG-11, WideResNet-18-4, WideResNet-18-2, GoogleNet

Dataset	Ensemble	Single Model	Single Model Control	Ensemble Control
CIFAR10 (InD)	93.44%	$94.32 \pm 0.1495\%$	$93.93 \pm 0.0804\%$	93.68%
CINIC10 (OOD)	67.88%	$68.84 \pm 0.3936\%$	$68.59 \pm 0.2762\%$	69.10%

Table 9: **Corresponding accuracies for models compared on CIFAR10/CIFAR10.1.** Architectures of models left to right: ResNet18, WideResNet18-4, WideResNet18, VGG-11

Dataset	Ensemble	Single Model	Single Model Control	Ensemble Control
CIFAR10 (InD)	94.26%	$94.28 \pm 0.1430\%$	$92.77 \pm 0.1839\%$	93.72%
CIFAR10.1 (OOD)	86.10%	$86.26 \pm 0.1727\%$	$84.61 \pm 0.4242\%$	84.12%