Combining Ensembles of Multi-Input Multi-Output Subnetworks and Data Augmentation Does not Harm Your Calibration

Çağrı Demir[1], Simon Caton[2], and Pierpaolo Dondio[1]

[1] School of Computer Science, Technological University Dublin, Ireland
cagri.a.demir@mytudublin.ie, pierpaolo.dondio@tudublin.ie
[2] School of Computer Science, University College Dublin, Ireland
simon.caton@ucd.ie

Abstract. Ensembling neural network models is a common practice to increase model calibration and robustness. Likewise, data augmentation is a set of techniques used to enhance model calibration and robustness by introducing invariant feature transformations. However, the total effect of combining two methods is not well researched. There are contradicting results presented in the literature showing that combining some ensembling methods and data augmentation can result miss-calibrated models. In this paper, we aim to show that data augmentation does not degrade model calibration for ensembles of multi-input-multi-output subnetworks. We find that combining ensembles of multi-input multi-output subnetworks with data augmentation increases accuracy without harming model calibration. Moreover, combining subnetwork ensembles with data augmentation also helps to achieve better uncertainty estimates. We designed and performed a factorial experiment consisting of 3 factors; data sets (Cifar-10, Cifar-100, Tiny ImageNet), ensembling frameworks (MIMO, Linear-MixMo, and Cut-MixMo), and data augmentation methods (MixUp and CutMix).

Keywords: Ensembles · Uncertainty Estimates · Calibration

1 Introduction

Deep learning models are starting to be used widely in safety-critical tasks such as autonomous driving [1] and medical applications. However, the data that models are trained and tested on can be different to the data used when these models are deployed in real-world scenarios. In such situations, these models need to be well-calibrated [8]. Both ensembling and data augmentation techniques have been shown to improve calibration, robustness, and model performance [9,13,17]. However, we still do not fully understand the effects (positive or negative) of combining ensembles with data augmentation methods.

Even simple averaging of the predictions can help reduce individual model misclassifications and other errors. There are different methods for ensembling models which have been shown to be effective in improving accuracy and robustness while not changing the total number of parameters significantly. Among others, Ensembles of multi-input multi-output subnetworks (Subnetwork Ensembles), BatchEnsemble [19] and its variants, and MC-dropout [6] are examples of such efficient ensembling methods [18]. The idea behind training subnetworks comes from sparsity. Recent deep learning models have millions of parameters. The overparametrization of deep learning models leads to the lottery ticket hypothesis [5] and model pruning methods [14]. Instead of pruning a model to get a subnetwork, Subnetwork Ensembles models take advantage of available neurons and overparametrizaton with little structural changes turning a single network into an ensemble of subnetworks. This method enables the generation of ensembles while increasing the total number of parameters by less than 1%. However, training such a model and ensuring independent subnetworks while sharing the main network's parameters with no explicit structural difference is a challenge.

Data augmentation methods encompass a diverse set of methods from basic geometric transformations of images to utilization of GANs [17]. These techniques try to emulate the distribution mismatch between the training and test data by increasing diversity among training images. Increasing the quality and quantity of image datasets helps to reduce neural networks' errors stemming from overconfidence. Consequently, models using data augmentation are less prone to over-fitting and have better generalization capability [12]. Almost all state-of-art vision models use one or a few data augmentation approaches.

In theory, data augmentation is orthogonal to ensembling [9,18]. Both ensembling and data augmentation increase accuracy, generalizability, and calibration. However, one can not directly combine ensembling and data augmentation without further analysis. The findings analyzing the interaction between ensembling and data augmentation are mixed in the literature. [18] shows how combining three ensembling methods (BatchEnsemble, MC-Dropout, and Deep Ensembles) with two Data Augmentation methods (Mixup and Augmix) without structural change on the said methods can harm the calibration of the model. However, [16] states that their findings do not confirm the pathology between ensembling and data augmentation, but that combining the two methods increases calibration.

In this paper, we try to clarify this conflict in the literature combining MIMO and MixMo neural networks (Subnetwork Ensembles) with data augmentation, and illustrate this combination does not harm model calibration while increasing accuracy. Moreover, ensemble and data augmentation combination also helps to achieve better uncertainty estimates. We confirmed this behavior across 3 different Subnetwork Ensembles frameworks and two data augmentation methods on three datasets. We also test all models on corrupted Cifar-10 and Cifar-100 datasets and find consistent results in the presence of corrupted data.

2 Related Work

2.1 Ensembles

Ensembling is a technique that takes advantage of diversity among different models to improve their combined performance [4]. Even simple ensembling (averaging predictions of randomly initialized neural networks) outperforms more complicated models. [13] show that deep ensembles trained independently improve both accuracy and calibration. However, there are many approaches for "ensembling" with BatchEnsemble, MC-Dropout, and Deep Subnetworks suggested in the literature. In this paper, we focus on Subnetwork Ensembles models.

Subnetwork Ensembles: Subnetwork Ensembles frameworks are based on the idea of sparsity. Recent neural network models are overparametrized. This leads to distilling and pruning methods to get a smaller network ("subnetwork") of the original network without sacrificing performance. Subnetwork Ensembles take advantage of these "free" subnetworks and utilizes them as an ensemble of networks to improve model performances. Recently several Subnetwork Ensembles frameworks have been proposed in the literature. MIMO and MixMo frameworks are based on this idea: training subnetworks that independently learn the task while utilizing a single model's capacity. The most distinctive feature of Subnetwork Ensembles is that these models take multiple inputs and predict them simultaneously. This structure allows them to flexibly exploit the base model's capacity stemming from overparametrization. However, how to train models under Subnetwork Ensembles frameworks and combine the inputs into a shared representation are still active areas of research.

MIMO: In MIMO (Figure 1), the network takes M inputs and outputs M outputs (predictions) where M is the number of desired subnetworks. MIMO requires only two changes: input layer takes M images which are simply stacked and output layer has M prediction vectors instead of a single one. In this sense, MIMO uses channel-wise concatenation in pixels for the inputs. These inputs are independently sampled from the training set and require no preprocessing. The base network is trained to predict matching images simultaneously. Each subnetwork learns to disregard features from other images. This ensures independence of subnetworks. The loss is calculated according to corresponding labels. During testing, the same input is repeated M times, and the outputs are averaged to get the final prediction. Clearly, MIMO does not need the neural network to have large structural changes. In terms of network structure, it is enough to change the first convolutional and last dense layers.

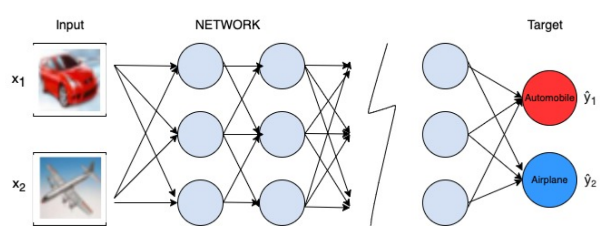

Fig. 1. MIMO framework with $M = 2$. The network receives 2 input images, stacks them and outputs a prediction for each image. All subnetworks share the same base network. At test time, the same input is repeated M times and predictions are averaged to obtain the final prediction.

MixMo: MixMo (Figure 2) has a similar setting to MIMO but instead of channel-wise concatenation of images in pixels, it first encodes each image and then employs a mixing block to combine inputs [16]. Inspired by mixing data augmentation methods, MixMo uses a generalized multi-input mixing block to combine inputs. In this regard, MixMo can be seen as a generalized form of MIMO. Using identity encoding layers and choosing channel-wise concatenation turns the MixMo framework into MIMO. However, the mixing block is not limited to any specific augmentation method; changing the mixing block results in a different framework. Following [16], we use MixUp and CutMix (see Figure 3) to mix input images, which we refer to as Linear-MixMo and Cut-MixMo.

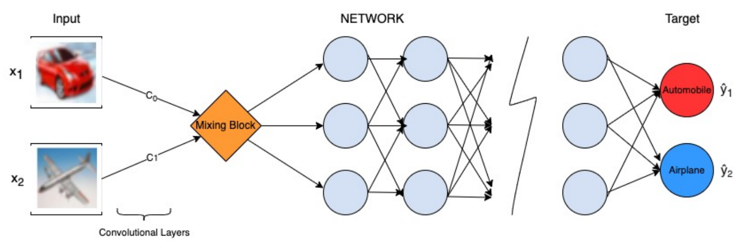

Fig. 2. MixMo framework with $M = 2$. The network receives 2 input images, encodes them with convolutional layers, mixes them according to the mixing operation (CutMix or MixUp) and outputs a prediction for each image. All subnetworks share the same base network. At test time, the same input is repeated M times and predictions are averaged to obtain the final prediction.

2.2 Data Augmentation

Data Augmentation (DA) increases the training data by introducing small perturbations or transformations (Figure 3). So models can be trained on more data. DA helps capture invariant feature transformations and is also used to simulate out-of-distribution data. Therefore, models utilizing DA tend to have better calibration and accuracy resulting in a large uptake of DA in the literature.

MixUp: MixUp is a simple data augmentation method which linearly interpolates pixels while manipulating the labels at the same time. The idea behind MixUp is that linear interpolations of feature vectors should lead to linear interpolations of target labels [22]. By doing so, MixUp extends training distribution. Given two random samples from training data (x_i, y_i) and (x_j, y_j), when Mixup is applied, we get $(\tilde{x}, \tilde{y})$ by:

$$\begin{aligned} \tilde{x} &= \lambda x_i + (1-\lambda)x_j \\ \tilde{y} &= \lambda y_i + (1-\lambda)y_j \end{aligned} \tag{1}$$

where λ is sampled from uniform distribution $\in [0, 1]$.

CutMix: CutMix creates new images by cutting patches from images and pasting them among training images. CutMix also mixes the true labels proportional to the area of the patches while patching. So a new training sample $(\tilde{x}, \tilde{y})$ is generated by combining two training samples $((x_a), (x_b))$ and $((x_b), (y_b))$. The combining operations are [20]:

$$\begin{aligned}\tilde{x} &= \mathbf{M} \odot x_a + (\mathbf{1} - \mathbf{M}) \odot x_b \\ \tilde{y} &= \lambda y_a + (1 - \lambda) y_b\end{aligned} \tag{2}$$

where M denotes a binary mask indicating where to drop out and fill in from two images, 1 is a binary mask filled with ones, and $\odot$ is element-wise multiplication. Like MixUp, λ is sampled from the uniform distribution $(0, 1)$.

Fig. 3. Common data augmentation methods [16]

2.3 Summary

Effects of ensembling and data augmentations on image classification tasks are well studied in the literature. However, we observe limited knowledge and guidance on what would be the total effect when these two seemingly orthogonal methods are combined. Being one of the recent ensembling strategies, Subnetwork Ensembles achieve ensembling by fitting diverse subnetworks inside a single base network. In this paper, we seek to provide some clarity on the effects of combining Subnetwork Ensembles with data augmentation methods and whether this improves model accuracy without harming model calibration.

3 Methodology

3.1 Experimental Design

This paper seeks to understand the impact of combining Subnetwork Ensembles with data augmentation. Ensembling and data augmentation are thought to be independent of each other [9,18] while both methods are used to enhance model performance. We try to verify [18]'s hypothesis on ensembling and data augmentation pathology. To do this, we perform a structured 3 x 3 x 2 factorial experimental design consisting of 3 factors; data sets (3), Subnetwork Ensembles frameworks (3), and data augmentation methods (2).

We trained all models on the Cifar-10, Cifar-100, and Tiny ImageNet datasets. Cifar-10 and Cifar-100 datasets both have 60k images (50k training and 10k test images) and 10 and 100 classes respectively. To further push the models we use Tiny ImageNet. Tiny ImageNet [2] is a downsampled variant of ImageNet as an alternative to the Cifar datasets with 64x64 pixels and with 100k total images and 200 classes (500 training, 50 validation, and 50 test images per class).

As Subnetwork Ensembles frameworks, we utilized Multi-input Multi-output (MIMO) [9] and two variants of MixMo [16] (Linear-MixMo and Cut-MixMo) as explained in section 2. [9] introduces input repetition and batch repetition during training. Input repetition helps subnetworks share the same features but degrades diversity among subnetworks. Following the MixMo paper ([16]), we don't utilize input repetition. Batch repetition has a regularization effect on the network training. MIMO finds batch repetition value $b = 4$ is optimal, and MixMo also uses $b = 4$. Hence we do the same. One of the core components of the subnetwork frameworks is the number of subnetworks. Since the original network's capacity is limited, as the number of total subnetworks increases, after an optimal number of subnetworks, the performance of the network decreases. Both MIMO and MixMo find that the optimal number of subnetworks is between 2 and 4 for the base models and datasets we utilized. Moreover, the number of subnetworks also increases the training time. We choose the number of subnetworks ($M = 2$) for all models. We follow the original papers for learning rate, optimization algorithm, and batch size.

To combine with Subnetwork Ensembles frameworks, we chose two common data augmentation methods: MixUp and CutMix. We go beyond simple data augmentations like flipping, rotation, pixel padding and use stronger augmentations. Indeed, our data augmentation methods fall into Mixed Sample Data Augmentation notion which basically manipulates both images and targets and creates virtual samples $((x_new), (y_new))$ given two pairs of input images $((x_i), (y_i))$ and $((x_j), (y_j))$ (see: section 2). Data augmentations are performed during training with the probability of 0.5 that a new training sample is generated.

Setting aside effects on performance, MIMO and MixMo frameworks can utilize almost all neural network models as base models, with the ResNet family as one of the most commonly used. Wide ResNets are known to have more sparsity than the original ResNets, and this helps Subnetwork Ensembles frameworks to better exploit its capacity. Following the original papers [9,16], all our models are based on a ResNet model, as they are sufficiently parameterized to enable good performance for subnetwork models. For Cifar-10 and Cifar-100, the base model is a WideResNet 28-10 (36.6 million parameters) [21] and for TinyImageNet the base model is PreActResNet-18 (11.2 million parameters) [10].

Neural networks encounter a dramatic decrease in their performance when they are tested against out-of-distribution data. After training all models with the matching framework, in addition to IID test sets, we tested all models on corrupted Cifar-10 and Cifar-100 test sets [11]. Images in this dataset are perturbed with 19 different common corruption types (e.g. added blur, compression artifacts, frost effects etc.) at 5 different severity levels. Thus, the Cifar-10 or Cifar-100 test set has 19x5=95 different unseen variations emulating out-of-distribution data. A model which improves performance on this should indicate general robustness gain and better calibration [11].

3.2 Performance Metrics: Calibration and Uncertainty Estimates

Calibration is a notion which measures how a model's predictions match the empirical frequency of the true probabilities [3]. We say that a model is well calibrated when a prediction of a class with confidence p is correct p% of the time. A model can have high accuracy yet be a miss-calibrated one. That is calibration and accuracy are two distinct phenomena. Measuring the predictive uncertainty estimates and how well a model is calibrated is a challenging task since the ground truth is not known. Therefore, we utilize two different metrics to measure the calibration: Expected Calibration Error (ECE) and Negative Log-Likelihood (NLL). We also use corrupted Cifar test sets to represent out-of-distribution examples to evaluate model calibration from a domain shift perspective.

By binning the predictions to M equally-spaced intervals and taking a weighted-average of each bins' accuracy, Expected Calibration Error (ECE) [15] measures the absolute difference between accuracy and predictive confidence, is widely used in the literature, and defined as follows:

$$\mathrm{ECE} = \sum_{m=1}^{M} \frac{|B_m|}{n} |\mathrm{acc}(B_m) - \mathrm{conf}(B_m)| \tag{3}$$

where $acc(B_m)$ is the average probability of the predicted and true class for the bin m and $conf(B_m)$ is the average confidence within (B_m).

Negative log-likelihood (NLL) is a proper scoring rule [13]. Scoring rules measure the quality of predictive uncertainty and rewards better calibrated predictions [7]. So maximizing likelihood (minimizing NLL) increases calibration. Given a probabilistic model π and n samples, NLL is defined as:

$$\mathcal{L} = -\sum_{i=1}^{n} \log(\hat{\pi}(y_i|x_i)) \tag{4}$$

4 Evaluation

After setting the experimental design and training all models, we tested all models on the respective test sets. We grouped our results for the metrics we track according to datasets. Moreover, we tested all models on corrupted Cifar-10 and Cifar-100. We report the accuracy metrics as well as the ECE and NLL metrics as discussed in section 3, which are averaged over 3 independent runs.

4.1 Results on Cifar-10/100 and TinyImageNet

Table 1 reports all model results tested on Cifar-10. Subnetwork Ensembles frameworks show a performance boost in terms of accuracy compared to the base models. They also improve calibration (lower ECE) and have better uncertainty estimates (lower NLL). When MIMO and MixMo are trained with MixUp and CutMix, model performance across all three metrics also increases. That is when ensemble models are combined with data augmentation, they better estimate uncertainty (lower NLL) and are better calibrated (lower ECE).

Table 2 reports results for models trained and tested on Cifar-100. Improvement in the metrics for Cifar-100 is similar to Cifar-10. Combining MixUp or CutMix with one of MIMO or MixMo makes all models more performant (higher accuracy) and better calibrated (lower NLL). Combining ensemble models with data augmentation methods results in performance gains across all metrics.

Table 3 reports results for models trained and tested on Tiny ImageNet. We see that results still have the general tendency to be improved when ensembling

Model	Data Augmentation	Accuracy(↑)	NLL(↓)	ECE(↓)
Base Model	–	96.31%	0.141	0.020
Base Model	MixUp	97.00%	0.115	0.010
Base Model	CutMix	97.21%	0.108	0.015
MIMO	–	96.66%	0.136	0.019
MIMO	MixUp	97.31%	0.103	0.008
MIMO	CutMix	97.61%	0.092	0.013
Linear-MixMo	–	96.78%	0.110	0.018
Linear-MixMo	MixUp	97.32%	0.104	0.009
Linear-MixMo	CutMix	97.50%	0.101	0.009
Cut-MixMo	–	97.42%	0.084	0.012
Cut-MixMo	MixUp	97.60%	0.083	0.001
Cut-MixMo	CutMix	97.70%	0.082	0.001

Table 1. Performance results for WRN-28-10/CIFAR10.

Model	Data Augmentation	Accuracy(↑)	NLL(↓)	ECE(↓)
Base Model	–	81.47%	0.762	0.065
Base Model	MixUp	83.15%	0.673	0.016
Base Model	CutMix	83.74%	0.661	0.050
MIMO	–	82.74%	0.740	0.076
MIMO	MixUp	84.04%	0.637	0.025
MIMO	CutMix	85.37%	0.562	0.034
Linear-MixMo	–	82.53%	0.685	0.067
Linear-MixMo	MixUp	84.15%	0.629	0.019
Linear-MixMo	CutMix	85.24%	0.564	0.035
Cut-MixMo	–	85.32%	0.548	0.045
Cut-MixMo	MixUp	85.41%	0.540	0.024
Cut-MixMo	CutMix	85.59%	0.533	0.019

Table 2. Performance results for WRN-28-10/CIFAR100.

is combined with data augmentation(s). Both MIMO and MixMo models have higher accuracy and lower calibration error when one of the data augmentations of MixUp and CutMix is added to the training. However, none of these combined models can outperform the base model combined with only CutMix in terms of accuracy. On the other hand, combining the base model with CutMix or MixUp outputs the worst calibrated models. This stands as an interesting case for data augmentation effects on model accuracy and calibration.

The test metrics on all three datasets imply that combining Subnetwork Ensembles with data augmentation improves accuracy, lowers NLL, and lowers ECE, i.e. combining them results in better performance and more calibrated models. This behavior is consistent across all combinations of Subnetwork Ensembles and data augmentations. Furthermore, combining Cut-MixMo and CutMix tends to result in the highest performance and most robust model.

Model	Data Augmentation	Accuracy(↑)	NLL(↓)	ECE(↓)
Base Model	–	62.56%	1.53	0.100
Base Model	MixUp	63.74%	1.62	0.121
Base Model	CutMix	65.09%	1.58	0.119
MIMO	–	62.40%	1.60	0.102
MIMO	MixUp	62.70%	1.54	0.093
MIMO	CutMix	64.50%	1.52	0.091
Linear-MixMo	–	61.58%	1.61	0.109
Linear-MixMo	MixUp	62.90%	1.5	0.092
Linear-MixMo	CutMix	63.78%	1.48	0.089
Cut-MixMo	–	62.91%	1.51	0.101
Cut-MixMo	MixUp	63.40%	1.49	0.088
Cut-MixMo	CutMix	64.44%	1.48	0.088

Table 3. Performance results for PreActResNet-18/Tiny ImageNet.

4.2 Models against image corruptions

Table 4 and Table 5 report results when all models are tested against corrupted Cifar datasets. Clearly, compared to IID test sets (uncorrupted), performances of all models on three metrics degrade. However, still, ensemble models with data augmentations are more calibrated than models without data augmentations.

When compared to the base model, Subnetwork Ensembles improve model performance. Linear-MixMo, on the other hand, outperforms Cut-MixMo and MIMO for all three metrics, in contrast to uncorrupted test sets. Using a data augmentation method on a corrupted Cifar-10 test set improves performance and calibration. All models combined with a data augmentation have higher accuracy and lower NLL and ECE. Applying MixUp improves performance metrics for all models more so than applying CutMix. Using only ensemble models or data augmentation on a corrupted Cifar-100 test set yields higher accuracy and lower NLL and ECE. Combining one of the MIMO or MixMo variants further improves performance. Unlike corrupted Cifar-10, CutMix performs marginally better than MixUp when used on top of the base model or in combination with one of the Subnetwork Ensembles.

To summarize, as in the case for uncorrupted test sets, utilizing Subnetwork Ensembles or data augmentations alone still enhances accuracy and decreases NLL and ECE. The best "combination" of Subnetwork Ensembles and data augmentation for "any" dataset is not clear but combining Subnetwork Ensembles with MixUp or CutMix almost always helps models further improve both model accuracy and calibration (lower ECE). This implies combining Subnetwork Ensembles with data augmentations will help model performance and calibration in the presence of out-of-distribution data. The main takeaway of the experiments with corrupted Cifar datasets is that combining Subnetwork Ensembles with data augmentation improves calibration in line with accuracy and vice versa. Finally, Figure 4 shows how ECE changes across each model family for all datasets. Clearly, adding an ensemble framework or data augmentation method improves model calibration. Moreover, combining Subnetwork Ensembles with data augmentation further improves model calibration measured via ECE.

Model	Data Augmentation	Accuracy(↑)	NLL(↓)	ECE(↓)
Base Model	–	76.77%	1.03	0.148
Base Model	MixUp	82.68%	0.64	0.058
Base Model	CutMix	77.09%	1.01	0.147
MIMO	–	77.06%	1.15	0.158
MIMO	MixUp	82.47%	0.62	0.057
MIMO	CutMix	78.41%	1.10	0.144
Linear-MixMo	–	80.18%	0.85	0.123
Linear-MixMo	MixUp	85.08%	0.51	0.032
Linear-MixMo	CutMix	79.08%	0.96	0.126
Cut-MixMo	–	79.36%	0.86	0.117
Cut-MixMo	MixUp	82.62%	0.60	0.041
Cut-MixMo	CutMix	79.45%	0.78	0.107

Table 4. Performance results for WRN-28-10/CIFAR10-corrupted.

Model	Data Augmentation	Accuracy(↑)	NLL(↓)	ECE(↓)
Base Model	–	51.40%	2.70	0.239
Base Model	MixUp	66.30%	1.42	0.180
Base Model	CutMix	67.80%	1.38	0.132
MIMO	–	53.70%	2.66	0.129
MIMO	MixUp	69.80%	1.38	0.116
MIMO	CutMix	70.10%	1.24	0.090
Linear-MixMo	–	55.60%	2.33	0.13
Linear-MixMo	MixUp	69.10%	1.39	0.115
Linear-MixMo	CutMix	70.40%	1.22	0.110
Cut-MixMo	–	57.00%	2.04	0.128
Cut-MixMo	MixUp	70.60%	1.27	0.106
Cut-MixMo	CutMix	71.10%	1.16	0.088

Table 5. Performance results for WRN-28-10/CIFAR100-corrupted.

5 Conclusion

Our experiments have illustrated that using Subnetwork Ensembles for data augmentation alone improves model calibration and robustness. More importantly, we find that combining Subnetwork Ensembles with MixUp or CutMix improves accuracy while not harming model calibration. Thus, adding some clarity to the literature on this point, as we did not observe any trade-off between ensembling and data augmentation for Subnetwork Ensembles. Rather, in our experiments, we observed that combining Subnetwork Ensembles and data augmentation improved calibration and uncertainty estimates. Our experiments with benchmark

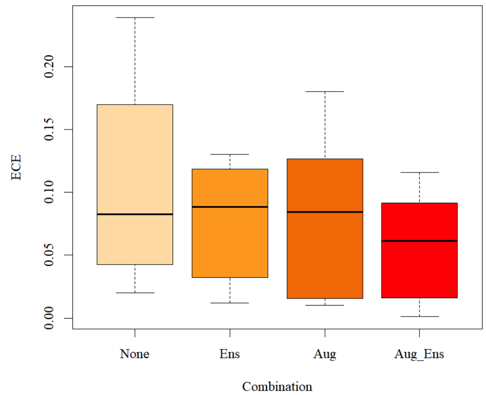

Fig. 4. ECE vs models across all datasets. *None* all models without data augmentation, *Ens* Subnetwork Ensembles models without data augmentation, *Aug* models using data augmentation, *Aug_Ens* Subnetwork Ensembles models using data augmentation.

corrupted datasets showed how the findings are also robust with respect to corruption since the minimum values for ECE and the NLL were obtained when both data augmentation and Subnetwork Ensembles were used.

Hence, combining Subnetwork Ensembles with data augmentation methods for image classification tasks helps to improve performance without sacrificing calibration. This situation signals a divergence on the effects of combining different methods for ensembling with data augmentation. Models trying to boost performance should consider this discrepancy. Exploring this behavior divergence (as future research) among ensembling methods when combined with data augmentation could yield a better understanding of seemingly uncorrelated methods.

Acknowledgments

This publication has emanated from research conducted with the financial support of Science Foundation Ireland under Grant number 18/CRT/6183. For the purpose of Open Access, the author has applied a CC BY public copyright licence to any Author Accepted Manuscript version arising from this submission. This work has been carried out using the ResearchIT Sonic cluster which was funded by UCD IT Services and the Research Office.

References

1. Bojarski, M., Del Testa, D., Dworakowski, D., et al.: End to end learning for self-driving cars. ArXiv (2016), `http://arxiv.org/abs/1604.07316`
2. Chrabaszcz, P., Loshchilov, I., Hutter, F.: A downsampled variant of imagenet as an alternative to the cifar datasets. ArXiv (2017), `https://arxiv.org/abs/1707.08819`
3. Degroot, M.H., Fienberg, S.E.: The comparison and evaluation of forecasters. The Statistician **32**, 12–22 (1983)
4. Dietterich, T.G.: Ensemble methods in machine learning. In: Multiple Classifier Systems. pp. 1–15. Springer Berlin Heidelberg, Berlin, Heidelberg (2000)

5. Frankle, J., Carbin, M.: The lottery ticket hypothesis: Finding sparse, trainable neural networks. In: International Conference on Learning Representations (2019)
6. Gal, Y., Ghahramani, Z.: Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In: Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48. ICML'16 (2016)
7. Gneiting, T., Raftery, A.E.: Strictly proper scoring rules, prediction, and estimation. Journal of the American Statistical Association **102**, 359–378 (2007)
8. Guo, C., Pleiss, G., Sun, Y., Weinberger, K.Q.: On calibration of modern neural networks. In: Proceedings of the 34th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 70, pp. 1321–1330 (2017)
9. Havasi, M., Jenatton, R., Fort, S., et al.: Training independent subnetworks for robust prediction. In: International Conference on Learning Representations (2021)
10. He, K., Zhang, X., Ren, S., et al.: Identity mappings in deep residual networks. In: Computer Vision – ECCV 2016 (2016)
11. Hendrycks, D., Dietterich, T.: Benchmarking neural network robustness to common corruptions and perturbations. Proceedings of the International Conference on Learning Representations (2019)
12. Hendrycks, D., Mu, N., Cubuk, E.D., et al.: Augmix: A simple method to improve robustness and uncertainty under data shift. In: International Conference on Learning Representations (2020)
13. Lakshminarayanan, B., Pritzel, A., Blundell, C.: Simple and scalable predictive uncertainty estimation using deep ensembles. In: Advances in Neural Information Processing Systems. vol. 30. Curran Associates, Inc. (2017)
14. Li, H., Kadav, A., Durdanovic, I., Samet, H., Graf, H.P.: Pruning filters for efficient convnets. In: 5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings. OpenReview.net (2017)
15. Naeini, M.P., Cooper, G.F., Hauskrecht, M.: Obtaining well calibrated probabilities using bayesian binning. In: Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence. p. 2901–2907 (2015)
16. Rame, A., Sun, R., Cord, M.: MixMo: Mixing multiple inputs for multiple outputs via deep subnetworks. In: ICCV (2021)
17. Shorten, C., Khoshgoftaar, T.M.: A survey on image data augmentation for deep learning. Journal of Big Data **6**, 1–48 (2019)
18. Wen, Y., Jerfel, G., Muller, R., et al.: Combining ensembles and data augmentation can harm your calibration. In: International Conference on Learning Representations (2021)
19. Wen, Y., Tran, D., Ba, J.: Batchensemble: an alternative approach to efficient ensemble and lifelong learning. In: International Conference on Learning Representations (2020)
20. Yun, S., Han, D., Oh, S.J., et al.: Cutmix: Regularization strategy to train strong classifiers with localizable features. In: International Conference on Computer Vision (ICCV) (2019)
21. Zagoruyko, S., Komodakis, N.: Wide residual networks. In: Proceedings of the British Machine Vision Conference (BMVC). pp. 87.1–87.12 (September 2016)
22. Zhang, H., Cisse, M., Dauphin, Y.N., Lopez-Paz, D.: mixup: Beyond empirical risk minimization. In: International Conference on Learning Representations (2018)