AUGMAX: ADVERSARIAL COMPOSITION OF RANDOM AUGMENTATIONS FOR ROBUST TRAINING REPRODUCIBILITY CHALLENGE

Shumeng Guo, Jess Newman, Hock Nien Gan
GitHub repository available at:
https://github.com/ShowmiGG/ReplicationChallenge2023

1 CORE CONCEPT OF THE PAPER

Data augmentation techniques help to artificially increase the size and diversity of the machine learning training dataset. The aim is to provide more information to the machine learning model about the underlying input data manifold and improve these models' generalisation performance. A core application of data augmentation is to prevent overfitting, which is one of the biggest problems in machine learning. AugMax is the sequel to previous research in data augmentation, RandAugment and AugMix. The motivation behind the research of AugMax is to improve the data augmentation technique to generate augmented images closer to the decision boundary of the data classes. Diversity and hardness are two important qualities the AugMax algorithm aims to balance. Diversity is the variance of the augmented images, and hardness is the difficulty to classify the augmented images. Image closer to the decision boundary make it harder for the model to classify the input images and helps to improve the robustness of the model. By increasing the variance of training samples, diversity makes it more likely the model will generalise well; unseen data points that may have been in an empty area in feature space before are now more likely to be similar to one of the diverse training data points. AugMax achieves a trade-off of the two by performing various random augments and then using adversarial training to obtain a worst-case mixture (ie. a hard-to-classify mixture) of the augmented images. First, some parallel augmentation chains are opened. Along all but one chain, a random number of augments is performed on the input image, each with a random severity; the original input image is maintained on the final chain. The weighted mixture of the augmented images is obtained by adding together each one with weights $\mathbf{w}*$, then this is mixed with the original chain with weights $1 - \mathbf{m}*$ and $\mathbf{m}*$. There is an inner maximisation step, where the adversarial attack attempts to maximise loss using these weight parameters, and an outer minimisation step where the overall loss of the classifier, plus an added consistency loss term, is minimised. The consistency loss is the JS divergence (similar to KL divergence but symmetric) between clean and augmented images, so this term essentially adds regularisation that ensures the augmented images are still visually recognisable as the original class. With this method, the authors claim to be able to outperform other methods on out-of-distribution robustness by 3.03%, 3.49%, 1.82% and 0.71% on CIFAR10-C, CIFAR100-C, Tiny ImageNet-C and ImageNet-C Wang et al. (2021). Another main contribution from this paper is the proposed Dual Batch-and-Instance Normalization (DuBIN) layer, which claims to help mitigate feature heterogeneity caused by augmentation. The author claims that this increases performance from naive AugMax. At each layer the features are put through a novel Dual Batch and Instance Normalisation (DuBIN) layer; the authors claim this disentangles heterogeneous features caused by augmentation. We will focus our efforts on reproducing the data that supports these claims on the advantages of AugMax and DuBIN.

2 EXPERIMENTAL METHODOLOGY

Our main goal is to test the claims made by the authors by reproducing their experiments. The key claims of the authors are that their adversarially trained AugMax module outperforms the similar, but fixed-weight AugMix module in terms of out-of-distribution robustness/distribution shift robustness. The claim that their DuBIN layer achieves yet higher performance also requires testing. In carrying out these experiments we also provide confirmation that data augmentation in general provides better generalisation to unseen data and out-of-distribution robustness, but this claim is already supported by evidence from several sources outside of the AugMax paper (Hendrycks et al. (2020), Cubuk et al. (2019)).

CIFAR10 is a core dataset used to evaluate the performance of the AugMax algorithm. We will attempt to reproduce the results by training models on these datasets, both with and without the AugMax data augmentation process. We also perform similar experiments on the TinyImageNet dataset. The robustness of the model against common corruption is evaluated using CIFAR10-C and TinyImageNet-C Hendrycks & Dietterich (2019). These datasets are produced by applying 15 types of corruption with five levels of severity for each type of image corruption (Fig 1). Models that perform well on these datasets are less vulnerable

to natural corruption and distribution shifts in real-world applications. The '-C' datasets are only used in evaluation so we can see how well it performs against completely unseen distribution shifts.

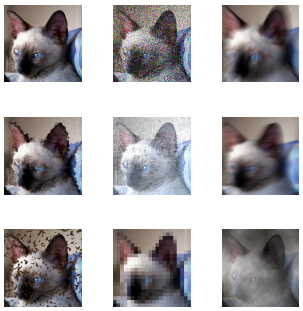

Figure 1: Some corruptions with severity 5 from the '-C' series of datasets

The original paper uses two evaluation metrics; standard accuracy (SA), the percentage of the correctly classified samples, and robustness accuracy (RA), the average classification accuracy over all 15 corruptions transformations. Robustness is an important measure to evaluate the models' performance in real-world scenarios where the model has to deal with different natural corruptions and distribution shifts. Achieving high robustness accuracy ensure that the trained model is less vulnerable to these corruptions and distribution shifts. The training time is also an important measure as it takes time for the AugMax algorithm to generate a large number of augmented images. It is important to evaluate the computational efficiency of the AugMax algorithm in comparison to other data augmentation techniques.

The AugMax authors also use mean corruption error (mCE), which is similar to robustness accuracy but corruptions are weighted by how difficult they are to overcome (determined by the number of errors a baseline model makes per corruption); this means a model is penalised less for misclassifying images with very severe corruptions.

To reproduce the paper's main claim of AugMax improving performance with out-of-distribution robustness, we would be looking for higher robustness accuracy with models trained with AugMax in our results. We also attempt to reproduce some of the ablation studies in the paper. These justify the claim that increased performance was indeed coming from the aspects they believed it was (as opposed to some unseen factor) by removing or adjusting certain parts of the framework. For example, if we see that individually the dual-batch normalisation layer provides a lower robustness accuracy than the DuBIN layer with instance normalisation, this is evidence of the author's claim they have created a novel solution to mitigating feature heterogeneity from data augmentation. We will attempt to reproduce this result, as well as an experiment where the authors were able to determine optimal AugMax hyperparameters by iterating through different values to see if these values are actually optimal.

3 Implementation Details

Thankfully, the authors of the AugMax paper have made a concerted effort to make their results reproducible; they provide the code for (most of) their paper in a GitHub repository, along with pre-trained models. The paper text also describes their experimental methodology in detail, with details such as optimisers and hyperparameters used included in the text. The only issue on this front is that some hyperparameters are missing or hard to find - for instance, the paper makes no mention of momentum in the optimiser, but their code suggests a value of 0.9 may have been used. Obviously, the pre-trained models do not prove much about their results as there is no way to know if they were actually trained as is claimed. We aim to recreate their experiments but use models we have trained from the ground up on the relevant datasets, using the framework they outline in the paper. The code they provide is written with the goal of utilising distributed computing to speed up the experimentation. For our reproducibility study, we have had to re-develop the code for our particular hardware, and to ensure the architecture actually follows the details in the paper. The original code base will be carefully used to support the development of our code for our experiments. Every effort has been made to reproduce the AugMax paper experiments as faithfully as possible. In cases where hardware limitations have made this restrictively difficult, we maintain the spirit of the experiments by comparing the robustness and accuracy of AugMax to other data augmentation frameworks with all else being equal.

Hyperparameter	**Value**
Optimiser	SGD
Epochs	50 (CIFAR), 100 (TIN)
Learning Rate	0.1
Batch Size	256
n (AugMax step no.)	10
α (AugMax step size)	0.1
λ (AugMax consistency loss factor)	10

Table 1: The hyperparameters and other implementation details used in training (unless otherwise stated).

ResNet18 is the main model we use the evaluate their claims and We trained 3 variants: a baseline model (no data augmentation), a model with AugMix-based data augmentation, and a model with AugMax-based data augmentation and the DuBIN normalisation layer. All of these are trained for 200 epochs with batch size 256 using an SGD optimiser with an initial learning rate of 0.1 and adjusted during training by Pytorch's cosine annealer scheduler. For the AugMax versions of these models, we use a PGD-based attack (Madry et al. (2019)) hyperparameters n = 10 to update the adversarial weights once every ten training updates, $\alpha = 0.1$ for the gradient ascent rate on the adversarial weights, and $\lambda = 10$ for the weighting for the consistency loss term. (The optimiser is minimising $L(\mathbf{x_{aug}}; \theta) + \lambda L_c(\mathbf{x}, \mathbf{x_{aug}})$).The CIFAR10 experiments were run on an NVIDIA RTX 3070.

Lambda	SA	RA
1	94.50	86.53
10	94.58	89.23
15	94.35	89.24

Table 2: Performance of AugMax-DuBIN CIFAR10 models with different consistency loss factor

Severity	SA	RA
1	94.81	89.33
3	94.58	89.16

Table 3: Performance metrics with different corruption severities

For Tiny ImageNet, we train for 100 epochs using a ResNet18 architecture. We do this repeatedly to get a baseline model, a model using AugMix, and a model using AugMax. We use a batch size of 32 and an SGD optimiser with a learning rate of 0.1. We reduce the learning rate to 0.09 at epoch 44 - the authors do this at epoch 100 but we observed the loss converged much, much earlier with no real gain in accuracy. We use the same AugMax hyperparameters, although we use a FAT-based adversarial attacker this time, which has an extra hyperparameter τ = 1. The Tiny ImageNet experiments were run on an NVIDIA GTX 1050.

During the evaluation, we generate the '-C' series test sets by applying the 15 possible corruptions at random with severity levels 1 and 3. To measure the training overhead imposed by AugMax, we take the average seconds per epoch for training a CIFAR10 -based model with AugMax, with AugMix, and without any data augmentation. For the ablation study, we see how replacing DuBIN with standard dual batch layers and instance normalisation affects performance on a CIFAR-10 ResNet18-based model with AugMax.

Due to hardware constraints adjustments to the hyperparameter settings were necessary such as reducing the number of epochs and the batch size. The training times can be seen in Table 7; it can be seen that AugMax has a significant overhead compared to the other methods - more significant than the AugMax paper found.

Method	CIFAR10-C
AugMax-DuBN	87.01
AugMax-DuBIN	89.23

Table 4: Performance of AugMax CIFAR10 ResNet18 model trained with different normalisation layers

We tested the performance of AugMax against different corruption severities: we found it was able to generalise well to several levels of corruption severity (Table 3).

Results for different CIFAR10 ResNet18-based models tested against the CIFAR10-C dataset are shown in Table 5. We can see, as expected, that the model trained with AugMax augmentation is the most robust to the corruption distribution shifts in the -C dataset. We found the baseline model was much less robust than the paper found. This suggests the author's claim that adding adversarial weights to AugMix does indeed improve the model's performance against 'hard' images (those that are very close to decision boundaries), as well as the claim that AugMax helps robustness in general, which can be seen by the increase in performance from the baseline model.

4 Discussion of findings

Results for different values of consistency loss factor are shown in 2. The idea of consistency loss is it keeps augmented images still looking like similar images visually to the clean version. We would expect a too-low value to lower performance, as the augmented images do not maintain as many of the features that represent the true class, and a too-high value to lower performance, as the model will be learning from cleaner images so will be less robust to very corrupt images. Unlike the paper, we did not see much difference between values of 10 and 15 for λ; perhaps an even higher value is required to enforce augmented images to be cleaner. Our experiments with DuBIN and DuBN layers support the author's claim

Method	SA	RA
Normal	94.38	10.90
AugMix	74.50	65.43
AugMax	94.58	89.23

Table 5: Performance metrics of CIFAR10 Resnet18 models with different data augmentation methods against CIFAR10-C

that the extra instance normalisation gives better performance for a given model by disentangling heterogeneous features.

Method	SA	RA	mCE
Normal	0.012	0.001	1
AugMix	0.3204	0.1243	0.9570
AugMax	0.5638	0.3272	0.8872

Table 6: Performance of the Tiny ImageNet trained ResNet18 models on Tiny ImageNet-C

The Tiny ImageNet results are shown in table 6. We found, similarly to the actual paper, that getting the Tiny ImageNet models beyond approximately 0.6 validation accuracy is very difficult; it quickly climbs to 0.5 before increasing very slowly. We found the training standard accuracy and robustness accuracy continue to climb, however, which may suggest an overfitting problem even though data augmentation is specifically supposed to prevent this. We did not find the same gains in accuracy by decreasing our learning rate that the authors did, but this could be because our decreases were timed differently so we may not have been at the optimal place in the loss landscape to decrease the learning rate. Our results confirmed that AugMax still performs better robustness-wise than other methods on the '-C' datasets all else being equal; this suggests the adversarial training and normalisation added to AugMax does indeed create models better at handling distribution shifts in the form of image corruption.

Our baseline performance in particular was much worse than even that which was found in the paper on the unseen corruptions. We also found, although this was not mentioned in the original paper, that AugMax models converged to a better validation accuracy much faster than other methods - some of the AugMix and baseline models were still nowhere near the accuracy of AugMax after the 100 epochs, but AugMax reached approximate convergence at about epoch 40.

5 Conclusions

Method	Training Time (sec/epoch)
Normal	104
AugMix	238
AugMax	16640

Table 7: Average training times for Tiny ImageNet for a ResNet18 on a GTX 1080.

Despite some oddities with the specific values of our results compared to the original paper, we found our results to reproduce similar conclusions for the most part. Models trained with AugMax data augmentation and DuBIN normalisation consistently showed better robustness accuracy than baseline methods when evaluated on a corruption dataset of distribution-shifted images. This suggests the authors are correct in claiming AugMax increases robustness to distributional shifts and better generalisation overall. Our ablation study looking at DuBN vs DuBIN performance shows that DuBIN has higher performance, so is indeed a novel contribution with useful applications to data augmentation; it's plausible this is due to the author's explanation of the extra instance normalisation mitigating the problem of very diverse features from data augmentation requiring larger models to fully capture by smoothing out the heterogeneity. We also found performance gains using AugMax over AugMix, suggesting the paper's other main claim of adversarially trained weights granting better hardness in the training set than the fixed weights of AugMix. The results, contributions, and claims of the AugMax paper are largely reproducible overall.

References

Ekin D. Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V. Le. Randaugment: Practical automated data augmentation with a reduced search space, 2019.

Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations, 2019.

Dan Hendrycks, Norman Mu, Ekin D. Cubuk, Barret Zoph, Justin Gilmer, and Balaji Lakshminarayanan. Augmix: A simple data processing method to improve robustness and uncertainty, 2020.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks, 2019.

Haotao Wang, Chaowei Xiao, Jean Kossaifi, Zhiding Yu, Anima Anandkumar, and Zhangyang Wang. Augmax: Adversarial composition of random augmentations for robust training. *CoRR*, abs/2110.13771, 2021. URL `https://arxiv.org/abs/2110.13771`.