On Misclassification-Aware Smoothing for Robustness and Uncertainty Calibration

Athanasios Tsiligkaridis [*1] **Theodoros Tsiligkaridis** [*2]

Abstract

Deep neural networks achieve high prediction accuracy when the train and test distributions coincide. In practice though, various types of corruptions can deviate from this setup and cause severe performance degradations. Few methods have been proposed to address generalization in presence of unforeseen domain shifts. In this paper, we propose a misclassification-aware Gaussian smoothing approach for improving robustness of image classifiers against a variety of corruptions while still maintaining high clean accuracy. With additional diverse data augmentations, we show that our method improves upon the state-of-the-art in robustness and uncertainty calibration on several image classification tasks.

1. Introduction

Deep neural networks are increasingly being used in computer vision and have achieved state-of-the-art performance on image classification (Krizhevsky et al., 2012; He et al., 2015; Huang et al., 2019). However, when the test distribution differs from the train distribution, performance can suffer as a result even for mild image corruptions and transformations (Hendrycks & Dietterich, 2019b). In fact, models have unrealistic behavior when faced with out-of-distribution inputs that arise from synthetic corruptions (Hendrycks & Dietterich, 2019b), spatial transformations (Engstrom et al., 2019), and data collection setups (Torralba & Efros, 2011; Recht et al., 2019). Although the mismatch between train and test distributions is common in practice, the problem has not been thoroughly studied yet. Thus, designing models that provide robustness to unforeseen corruptions or deviations from the train distribution is highly desirable.

[*]Equal contribution [1]Department of Electrical and Computer Engineering, Boston University, Boston, MA, USA [2]Artificial Intelligence Technology Group, MIT Lincoln Laboratory, Lexington, MA USA. Correspondence to: Athanasios Tsiligkaridis <atsili@bu.edu>.

Presented at the ICML 2021 Workshop on Uncertainty and Robustness in Deep Learning., Copyright 2021 by the author(s).

The natural approach to defending against a particular fixed distribution shift is to explicitly incorporate such data into the training process, see e.g. (Kang et al., 2019). However, this paradigm has drawbacks including over-fitting to one type of corruption (Geirhos et al., 2018), e.g., in (Kang et al., 2019) it was shown that ℓ_∞ robustness provides poor generalization to unforeseen attacks. Furthermore, the empirical study in (Chun et al., 2019) shows that several expensive methods improve robustness at the cost of lower clean accuracy and there are large trade-offs in corrupt and clean accuracies for a variety of regularization methods. Data augmentation policies have been proposed to increase clean accuracy (Cubuk et al., 2019) based on reinforcement learning but are computationally expensive. Recent work (Hendrycks et al., 2020) proposes randomized and diverse augmentations coupled with a consistency loss to improve robustness against corruptions while maintaining clean accuracy. However, the performance gap between clean and corrupt accuracy can be further improved.

In this work, we introduce a misclassification-aware consistency loss coupled with Gaussian noise regularization and a corresponding training algorithm. It is shown experimentally that using this approach in conjunction with diverse data augmentations achieves state-of-the-art (SOTA) generalization performance against a large variety of image corruptions for several image classification tasks and architectures.

2. Background and Related Work

We assume labeled data of the form $(x, y) \sim \mathcal{D}$ drawn from distribution $\mathcal{D}$. The labels y correspond to C classes. Neural network function $f_\theta(\cdot)$ maps inputs into logits, and θ are the model parameters. The softmax layer is used to map logits into class probability scores given by $p_c(x) = e^{f_{\theta,c}(x)} / \sum_l e^{f_{\theta,l}(x)}$.

Standard Training. The standard criterion for training deep neural networks is empirical risk minimization (ERM):

$$\min_\theta \mathop{\mathbb{E}}_{(x,y)\sim\mathcal{D}} [\mathcal{L}(f_\theta(x), y)] \tag{1}$$

where the loss is chosen to be the cross-entropy function $\mathcal{L}(f_\theta(x), y) = -y^T \log p_\theta(x)$. While training using the

criterion (1) yields high accuracy on clean test sets, the network generalization performance to a variety of data shifts may suffer.

Robust Training against ℓ_p adversary. Adversarial training (AT) (Madry et al., 2018) is one of the most effective defenses against L_p perturbations which minimizes the adversarial risk,

$$\min_{\theta} \mathop{\mathbb{E}}_{(x,y)\sim\mathcal{D}} \left[\max_{\|\delta\|_p \leq \epsilon} \mathcal{L}(f_\theta(x+\delta), y) \right] \tag{2}$$

During the training process, adversarial attacks are computed at inputs x that solve the inner maximization problem. The inner maximization may be solved iteratively using projected gradient descent (PGD) for norms $p \in \{2, \infty\}$, i.e., $\delta^{(k+1)} = \mathcal{P}_{B_p(\epsilon)}(\delta^{(k)} + \alpha \nabla_\delta \mathcal{L}(x+\delta^{(k)}, y))$ where $\mathcal{P}_{B_p(\epsilon)}(z) = \arg\min_{u \in B_p(\epsilon)} \|z-u\|_2^2$ is the orthogonal projection onto the constraint set. Another robust training approach that trades-off the natural and robust error using smoothing is TRADES (Zhang et al., 2019):

$$\min_{\theta} \mathop{\mathbb{E}}_{(x,y)\sim\mathcal{D}} [\mathcal{L}(f_\theta(x), y) + \lambda D(p(x;\theta) \parallel p(x+\delta';\theta))] \tag{3}$$

where D denotes the Kullback-Leibler divergence $D(p \parallel q) = \sum_c p_c \log \frac{p_c}{q_c}$ and δ' is the adversarial perturbation computed using PGD that solves the maximization problem $\max_{\|\delta\|_p \leq \epsilon} D(p(x;\theta) \parallel p(x+\delta;\theta))$. In the context of adversarial robustness, a random self-ensemble (RSE) method has been proposed (Liu et al., 2018) based on noise injection at the input layer and each layer of neural networks and was demonstrated to provide good levels of robustness against white-box attacks. Considering noise at the input layer only, the RSE training criterion is:

$$\min_{\theta} \mathop{\mathbb{E}}_{(x,y)\sim\mathcal{D}} \left[\mathop{\mathbb{E}}_{\delta\sim\mathcal{N}(0,\sigma^2 I)} \mathcal{L}\left(f_\theta(x+\delta), y\right) \right] \tag{4}$$

and predictions are ensembled at inference time as $\hat{y}(x) = \arg\max_c \frac{1}{n} \sum_{i=1}^{n} p_c(x+\delta_i)$ where $\delta_i \sim N(0, \sigma^2 I)$.

Robustness against Domain Shifts. A recently proposed data augmentation technique, AugMix (Hendrycks et al., 2020), was shown to achieve SOTA performance against unforeseen corruptions by enforcing a consistency loss coupled with a data augmentation scheme:

$$\min_{\theta} \mathop{\mathbb{E}}_{(x,y)\sim\mathcal{D}} \Big[\mathcal{L}\left(f_\theta(x), y\right) + \lambda JS(p_\theta(x); p_\theta(x_{a,1}); p_\theta(x_{a,2})) \Big] \tag{5}$$

where $JS(p_1; p_2; p_3) = \frac{1}{3}(D_{KL}(p_1 \parallel p_{\text{mix}}) + D_{KL}(p_2 \parallel p_{\text{mix}}) + D_{KL}(p_3 \parallel p_{\text{mix}}))$ is the Jensen-Shannon divergence, $p_{\text{mix}} = \frac{1}{3}(p_1 + p_2 + p_3)$, and $x_{a,1}, x_{a,2}$ are augmented variants of x formed by mixing composition chains consisting of a finite number of augmentation operations such as rotate, posterize, shear, translate, solarize, etc. Using diversity and randomness in choosing these operations and mixing weights at different levels of severity during training, this data augmentation method is empirically shown to significantly improve robustness against unforeseen corruptions in comparison to CutOut (DeVries & Taylor, 2017), MixUp (Zhang et al., 2017; Tokozume et al., 2018), CutMix (Yun et al., 2019), and AutoAugment (Cubuk et al., 2018) schemes.

3. Misclassification-Aware Gaussian Noise Training

We propose a training algorithm to improve generalization against a variety of corruptions while fitting in existing pipelines with minimal changes. The idea is to introduce a misclassification-aware consistency loss that embeds representations of clean data samples and diversified noise-corrupted versions similarly. Our proposed training criterion is:

$$\min_{\theta} \mathop{\mathbb{E}}_{(x,y)\sim\mathcal{D}} \Big[\mathcal{L}\left(f_\theta(x), y\right) + \lambda_1 [p_\theta(x)]_y \mathop{\mathbb{E}}_{\sigma\sim\mathcal{U}([0,\sigma_{\max}])} \mathop{\mathbb{E}}_{\delta\sim\mathcal{N}(0,\sigma^2)} D(p_\theta(x) \parallel p_\theta(x+\delta)) \Big] \tag{6}$$

The classification loss $\mathcal{L}$ in (6) is the standard term that maximizes accuracy on clean examples. Our regularizer forces clean examples and random noisy perturbed data to have similar output distributions. Weighting is also applied using the *true class confidence* level $[p_\theta(x)]_y$ which allows for the network training to regularize more on high-confidence correct predictions and focus more on the classification loss when the predictions are incorrect. To increase resiliency against a variety of Gaussian noise distributions, we diversify the perturbation statistics by choosing a random noise level σ uniformly in the range $[0, \sigma_{\max}]$ and then generating the random perturbation δ. Interestingly, training with diverse Gaussian noise augmentations coupled with the misclassification-aware weighting provides robustness not only against Gaussian noise but also against a variety of weather, blur, noise and digital corruptions, as evidenced in our results.

We call our approach Misclassification-Aware Gaussian Noise (MAGN) smoothing, and Algorithm 1 depicts our training procedure. A single stochastic draw is used to approximate the MAGN regularizer to maintain low computational complexity during training, while multiple stochastic draws of σ and corresponding δ could also be used to improve performance at the expense of increased training time.

Mathematical analysis of the effect of Gaussian noise regularization in the small noise regime is provided in the

Algorithm 1 MAGN pseudocode

Input: Training data $\{(x_i, y_i)\}$, Network f_θ, Training epochs T, Batch size $|B|$, learning rate schedule η_t, hyperparameters $(\lambda_1, \lambda_2, \sigma_{\max})$
Result: Trained network f_θ
for t=0 to $T-1$ **do**
  for each batch $(x, y) \sim \mathcal{D}$ **do**
    Generate $\sigma \sim \mathcal{U}(0, \sigma_{\max})$ for each batch example
    Generate $\delta \sim \mathcal{N}(0, \sigma^2 I)$ for each batch example
    $x_\delta = x + \delta$
    $\mathcal{L}_T(x_i, y_i) = \mathcal{L}(x_i, y_i) + \lambda_1 [p_\theta(x_i)]_{y_i} D(p_\theta(x_i) \parallel p_\theta(x_{i,\delta}))$
    $\theta = \theta - \eta_t \frac{1}{|B|} \sum_{i \in B} \nabla_\theta \mathcal{L}_T(x_i, y_i)$
  end for
end for

Supplementary Material section. This analysis is used to understand the effect of the MAGN regularizer on the local loss landscape and motivate our misclassification-aware modification.

4. Experimental Results

Datasets. The datasets used in this experimental results are CIFAR-10 and CIFAR-100, both containing color images of size $32 \times 32 \times 3$ spanned across $50,000$ train images and $10,000$ test images. Robustness against data shifts is measured by evaluating on CIFAR-10-C and CIFAR-100-C respectively (Hendrycks & Dietterich, 2019a). Each of these corrupted datasets contain a total of $M = 18$ corruptions at $J = 5$ severity levels. The 'gaussian noise' corruption is excluded from our evaluations.

Training Details. The MAGN training method is demonstrated on deep residual network architecture ResNet-18 (He et al., 2016), densely connected convolutional architecture DenseNet-121 (Huang et al., 2017), and Inception-V3 (Szegedy et al., 2015). The learning rate is started at 0.1 and decays every 50 epochs by a factor of 10. Pre-processing steps include standard random cropping, random horizontal flips, color jitter (0.25) and random rotation (2 degrees) prior to additional noise perturbations or histogram changes. Training was performed for 150 epochs using SGD with Nesterov momentum 0.9 and weight decay 0.0005. A noise standard deviation $\sigma = 0.1$ was chosen to achieve a high mCA experimentally for the RSE method with $n = 10$ test evaluations, and for the MAGN regularizer $\sigma_{\max} = 0.2$ was used for all experiments for consistency.

Baselines. The baseline methods considered include standard training, adversarial training (AT), tradeoff-inspired adversarial defense via surrogate-loss minimization (TRADES), random self-ensemble (RSE), and data augmentation method to improve robustness (AugMix). The adversarial models AT and TRADES were trained against an ℓ_∞ adversary using $\epsilon = 8/255$ and 7 PGD steps with step size $2.5\epsilon/7$ to allow sufficient exploration of the constraint set's boundary.

Performance Metrics. Classification performance on the clean dataset is measured using test accuracy, i.e., $\frac{1}{n}\sum_{i=1}^{n} \mathbb{1}_{\{y_i = \hat{y}_i\}}$. Robustness against domain shifts is measured using accuracy on a corrupted validation set for different severity levels j. For a specific corruption type $m \in \{1, \ldots, M\}$ and severity level $j \in \{1, \ldots, J\}$, let $A_{m,j}$ denote the corresponding accuracy. The *mean corruption accuracy (mCA)* is defined as: $A_M = \frac{1}{MJ}\sum_{m=1}^{M}\sum_{j=1}^{J} A_{m,j}$.

Classifier calibration refers to the problem of true empirical correct likelihood matching the predicted confidence metric. This leads to trustworthy probability estimates of the model predictions. Uncertainty calibration performance is measured using the root-mean-square (RMS) calibration. Consider a partition of n_B bins $\{B_i\}$ that correspong to increasing levels of confidence. Then, RMS calibration error, which measures the discrepancy between the empirical accuracy and the prediction confidence level, is computed as: $E_c = \sqrt{\sum_{i=1}^{n_B} \frac{|B_i|}{n} \left(\frac{1}{|B_i|} \sum_{j \in B_i} \mathbb{1}_{\{y_j = \hat{y}_j\}} - \frac{1}{|B_i|} \sum_{j \in B_i} c_j \right)^2}$.

Robustness Results. Classification accuracy results on the clean and corrupted CIFAR-10 test sets are shown in Table 1 for a variety of deep learning methods on the ResNet-18 architecture. In comparison to the 'Standard' baseline trained with (1), our misclassification-aware Gaussian noise smoothing (MAGN) achieves 8.5% mCA absolute improvement. Furthermore, when combined with AugMix, MAGN + AugMix achieves 13.2% mCA improvement over standard training, outperforming the previous SOTA method AugMix which obtained a smaller 9.9% improvement, adversarial training and random self-ensemble baselines by a significant margin. In particular, MAGN achieves the best corrupt accuracy for certain challenging noise corruptions such as shot noise and speckle noise for which AugMix lags behind. We further note that MAGN maintains a high clean accuracy, unlike the adversarial training methods AT and TRADES, and does not require ensembling at test time as in RSE. Including both the MAGN and AugMix consistency losses regularizes the model in complementary ways and improves generalization to a variety of corruptions.

Classification accuracy results on additional architectures are shown in Table 2 which continue to support that the combination of our MAGN approach with AugMix achieves SOTA performance on CIFAR-10-C and CIFAR-100-C with average mCA improvements of 14.5% and 16.7%, respectively, over the standard models.

Uncertainty Calibration Results. Calibration performance is measured on the clean and corrupted CIFAR-10

Corruption	Standard	AT	TRADES	RSE	AugMix	MAGN	MAGN+AugMix
Clean	**94.74**	85.71	85.12	91.00	94.26	93.19	94.66
Brightness	94.02	84.07	83.13	89.78	93.65	91.95	**94.09**
Contrast	84.66	51.22	42.95	63.55	90.36	70.89	**91.72**
Defocus Blur	81.70	81.42	78.68	85.55	92.40	85.55	**93.17**
Elastic Transform	84.00	79.89	77.01	84.18	89.25	85.84	**89.97**
Fog	90.68	65.63	56.98	72.86	90.31	80.25	**90.92**
Frost	79.83	79.74	76.70	86.91	86.42	89.05	**91.27**
Gaussian Blur	72.83	79.65	76.30	83.17	91.25	81.92	**92.06**
Glass Blur	53.24	79.66	76.03	81.11	73.98	81.48	**83.49**
Impulse Noise	57.35	74.92	72.95	83.87	81.88	89.58	**89.82**
JPEG Compression	79.33	83.61	82.49	88.74	87.26	**89.72**	88.91
Motion Blur	77.36	77.83	74.25	80.72	89.54	80.50	**90.72**
Pixelate	74.95	83.63	82.32	88.17	82.49	**88.90**	86.40
Saturate	**92.60**	81.73	81.54	87.17	91.85	90.31	92.56
Shot Noise	60.70	81.40	80.38	89.17	80.42	**91.45**	91.13
Snow	83.36	81.22	80.27	85.13	87.98	86.85	**90.11**
Spatter	85.97	80.99	79.79	85.35	91.09	87.82	**92.06**
Speckle Noise	64.35	80.90	79.64	88.93	81.38	**91.60**	91.26
Zoom Blur	77.58	80.85	76.97	84.21	91.01	84.30	**91.91**
mCA	77.47	78.24	75.46	83.80	87.36	85.99	**90.64**

Table 1. Classification accuracy on CIFAR-10-C across different corruptions for various methods on ResNet-18 architecture. The best accuracy is highlighted in bold for each row which corresponds to a specific corruption. MAGN + AugMix achieves the highest mCA score.

Architecture	Test Set	Standard	AugMix	MAGN+AugMix
ResNet-18	CIFAR-10	94.74	94.26	94.66
	CIFAR-10-C	77.47	87.36	**90.64**
DenseNet-121	CIFAR-10	93.84	94.55	94.54
	CIFAR-10-C	75.13	87.74	**89.63**
Inception-V3	CIFAR-10	94.30	95.72	95.87
	CIFAR-10-C	74.73	89.26	**90.61**
ResNet-18	CIFAR-100	76.16	74.96	75.94
	CIFAR-100-C	50.96	63.08	**66.82**
DenseNet-121	CIFAR-100	73.74	76.12	75.60
	CIFAR-100-C	47.35	63.73	**64.78**
Inception-V3	CIFAR-100	75.57	78.35	78.16
	CIFAR-100-C	51.45	66.76	**68.37**

Table 2. Classification accuracy on CIFAR-10/CIFAR-100 datasets and Mean Corrupted Accuracy (mCA) on CIFAR-10-C/CIFAR-100-C corrupted datasets for ResNet-18, DenseNet-121 and Inception-V3 architectures.

Architecture	Test Set	Standard	AugMix	MAGN+AugMix
ResNet-18	CIFAR-10	4.7	3.2	2.2
	CIFAR-10-C	17.7	1.7	**1.1**
DenseNet-121	CIFAR-10	6.4	3.0	4.1
	CIFAR-10-C	20.9	2.4	**1.9**
Inception-V3	CIFAR-10	5.3	1.5	2.0
	CIFAR-10-C	19.9	5.0	**3.3**
ResNet-18	CIFAR-100	8.4	4.3	1.8
	CIFAR-100-C	20.2	**1.8**	5.5
DenseNet-121	CIFAR-100	15.0	4.7	5.4
	CIFAR-100-C	30.9	2.1	**0.6**
Inception-V3	CIFAR-100	8.0	3.7	3.9
	CIFAR-100-C	19.7	**8.4**	8.7

Table 3. RMS calibration error (percentage) on CIFAR-10/CIFAR-10-C and CIFAR-100/CIFAR-100-C datasets on ResNet-18, DenseNet-121 and Inception-V3 architectures. MAGN + AugMix achieves a large reduction in RMSE calibration error in comparison to standard training.

validation sets on different model architectures as shown in Table 3. The combination of our MAGN regularizer with AugMix yields significant improvements in RMSE calibration errors in comparison to standard training. Specifically, it achieves an average absolute improvement of 17.4% and 18.6% in RMSE calibration error over standard training on the corrupted CIFAR-10-C and CIFAR-100-C validation sets, respectively.

Additional experimental results on the clean and corrupted CIFAR-100 datasets can be found in the Supplementary Material section.

5. Conclusion

In this paper, a regularization method for training robust deep learning classifiers is presented based on misclassification-aware Gaussian smoothing at different scales. We empirically show that combining this type of Gaussian noise smoothing with additional data augmentation mixing chains yields state-of-the-art robustness against unforeseen domain shifts, while also improving uncertainty calibration for different architectures and datasets. We hope this work encourages more research on improving generalization of classifiers against natural occurring corruptions.

References

Chun, S., Oh, S. J., Yun, S., Han, D., Choe, J., and Yoo, Y. An empirical evaluation on robustness and uncertainty of regularization methods. In *ICML Workshop on Uncertainty and Robustness in Deep Learning*, 2019.

Cubuk, E. D., Zoph, B., Mane, D., Vasudevan, V., and Le, Q. V. Autoaugment:learning augmentation policies from data. In *CVPR*, 2018.

Cubuk, E. D., Zoph, B., Mane, D., Vasudevan, V., and Le, Q. V. Autoaugment: Learning augmentation strategies from data. In *CVPR*, 2019.

DeVries, T. and Taylor, G. W. Improved regularization of convolutional neural networks with cutout. In *arXiv:1708.04552*, 2017.

Engstrom, L., Tran, B., Tsipras, D., Schmidt, L., and Madry, A. Exploring the landscape of spatial robustness. In *ICML*, 2019.

Geirhos, R., Temme, C. R. M., Rauber, J., Schutt, H. H., Bethge, M., and Wichmann, F. A. Generalisation in humans and deep neural networks. In *NeurIPS*, 2018.

He, K., Zhang, X., Ren, S., and Sun, J. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *ICCV*, 2015.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *CVPR*, 2016.

Hendrycks, D. and Dietterich, T. Benchmarking neural network robustness to common corruptions and perturbations. In *ICLR*, 2019a.

Hendrycks, D. and Dietterich, T. G. Benchmarking neural network robustness to common corruptions and surface variations. In *ICLR*, 2019b.

Hendrycks, D., Mu, N., Cubuk, E. D., Zoph, B., Gilmer, J., and Lakshminarayanan, B. Augmix: A simple data processing method to improve robustness and uncertainty. In *ICLR*, 2020.

Huang, G., Liu, Z., van der Maaten, L., and Weinberger, K. Q. Densely connected convolutional networks. In *CVPR*, 2017.

Huang, Y., Cheng, Y., Bapna, A., Firat, O., Chen, M. X., Chen, D., Lee, H., Ngiam, J., Le, Q. V., Wu, Y., and Chen, Z. Gpipe: Efficient training of giant neural networks using pipeline parallelism. In *NeurIPS*, 2019.

Kang, D., Sun, Y., Hendrycks, D., Brown, T., and Steinhardt, J. Testing robustness against unforeseen adversaries. In *arXiv pre-print arxiv:1908.08016*, 2019.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, 2012.

Kullback, S. *Information Theory and Statistics*. Dover, 1997.

Lin, C., Martens, J., Gowal, S., Krishnan, D., Dvijotham, K., Fawzi, A., De, S., Stanforth, R., and Kohli, P. Adversarial robustness through local linearization. In *NeurIPS*, 2019.

Liu, X., Cheng, M., Zhang, H., and Hsieh, C.-J. Towards robust neural networks via random self-ensemble. In *ECCV*, 2018.

Madry, A., Makelov, A., Schmidt, L., Tsipras, D., and Vladu, A. Towards deep learning models resistant to adversarial attacks. In *ICLR*, 2018.

Recht, B., Roelofs, R., Schmidt, L., and Shankar, V. Unbiased look at dataset bias. In *ICML*, 2019.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., and Wojna, Z. Rethinking the inception architecture for computer vision. In *arXiv:1512.00567v3*, 2015.

Tokozume, Y., Ushiku, Y., and Harada, T. Between-class learning for image classification. In *CVPR*, 2018.

Torralba, A. and Efros, A. A. Unbiased look at dataset bias. In *CVPR*, 2011.

Yun, S., Han, D., Oh, S. J., Chun, S., Choe, J., and Yoo, Y. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *ICCV*, 2019.

Zhang, H., Cisse, M., Dauphin, Y., and Lopez-Paz, D. mixup: Beyond empirical risk minimization. In *ICLR*, 2017.

Zhang, H., Yu, Y., Jiao, J., Xing, E. P., Ghaoui, L. E., and Jordan, M. I. Theoretically principled trade-off between robustness and accuracy. In *ICML*, 2019.

6. Supplementary Material

6.1. Data Corruption Visualizations

An example image along with its corrupted versions is shown in Figure 1; although humans might be able to recognize this as a boat, a deep learning image classifier trained in a standard manner will fail against most of these corruptions.

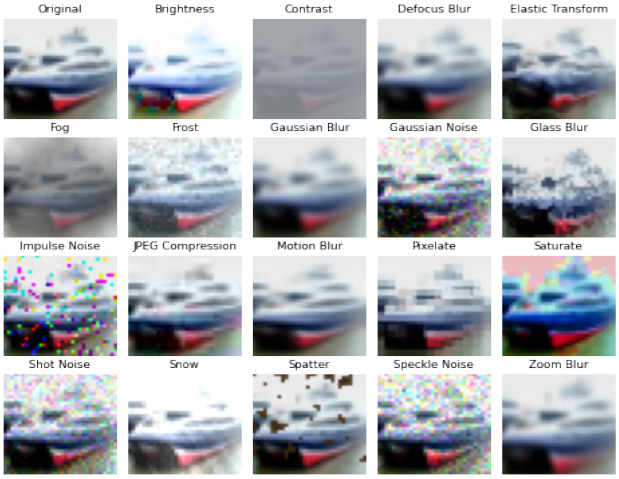

Figure 1. Example CIFAR-10 test image along with its corrupted versions. These corruptions are not available in the training process, and are only used for evaluation at inference time.

6.2. Analysis of Gaussian Noise Regularization

In this section, we analyze the effect of Gaussian noise regularization in the small-noise regime. Using this analysis, we obtain a relationship between loss curvature and Fisher information, which is further used to derive a bound on the local loss deviation in a neighborhood of x. This bound is used to understand the effect of the MAGN regularizer on the local loss landscape and motivate our misclassification-aware modification.

Using a second-order Taylor expansion on the KL divergence D, for small σ, we have (Kullback, 1997):

$$\begin{aligned}
&\mathop{\mathbb{E}}_{\delta\sim\mathcal{N}(0,\sigma^2 I)} D(p_\theta(x) \,\|\, p_\theta(x+\delta)) \\
&\quad\approx \mathop{\mathbb{E}}_{\delta\sim\mathcal{N}(0,\sigma^2 I)} \frac{1}{2}\delta^T G_\theta(x)\delta \\
&\quad= \mathop{\mathbb{E}}_{\delta\sim\mathcal{N}(0,\sigma^2 I)} \frac{1}{2}\operatorname{Tr}(G_\theta(x)\delta\delta^T) \\
&\quad= \frac{\sigma^2}{2}\operatorname{Tr}(G_\theta(x)) \qquad (7)
\end{aligned}$$

where we used $\mathbb{E}_{\delta\sim\mathcal{N}(0,\sigma^2 I)}[\delta\delta^T] = \sigma^2 I$, and $G_\theta(x)$ is the Fisher information matrix (FIM) given by:

$$G_\theta(x) = \sum_k [p_\theta(x)]_k \nabla_x \log[p_\theta(x)]_k (\nabla_x \log[p_\theta(x)]_k)^T \qquad (8)$$

Taking the outer expectation as in the first regularizer of (6) and using the approximation (7):

$$\begin{aligned}
&\mathop{\mathbb{E}}_{\sigma\sim\mathcal{U}([0,\sigma_{\max}])} \mathop{\mathbb{E}}_{\delta\sim\mathcal{N}(0,\sigma^2 I)} D(p_\theta(x) \,\|\, p_\theta(x+\delta)) \\
&\qquad\approx \mathop{\mathbb{E}}_{\sigma\sim\mathcal{U}([0,\sigma_{\max}])} \left[\frac{\sigma^2}{2}\right] \operatorname{Tr}(G_\theta(x)) \\
&\qquad= \frac{\sigma_{\max}^2}{6}\operatorname{Tr}(G_\theta(x)) \qquad (9)
\end{aligned}$$

The FIM has a strong connection to curvature; in fact for the case of cross-entropy, the Hessian matrix of the loss function is identical to the FIM (8), as the next proposition shows.

Proposition 1 *The following relation holds* $H(x) := \nabla_x^2 \mathcal{L}(f_\theta(x), y) = G_\theta(x)$.

Proof: From Appendix C in (Lin et al., 2019), the Hessian of the softmax cross-entropy function can be decomposed as:

$$H(x) = J^T(\operatorname{diag}(p) - pp^T)J \qquad (10)$$

where J denotes the Jacobian of the network function f_θ (logits) and p denotes the softmax probabilities. The Jacobian transposed is denoted as $J^T = [J_1^T, \ldots, J_K^T]$ with $J_k := \nabla_x f_k(x)^T$. Starting from the FIM in (8), we have:

$$\begin{aligned}
&G_\theta(x) = \sum_k p_k(\nabla_x \log p_k)(\nabla_x \log p_k)^T \\
&= \sum_k p_k(\nabla_x f_k - J^T p)(\nabla_x f_k - J^T p)^T \\
&= \sum_k p_k(\nabla_x f_k \nabla_x f_k^T - \nabla_x f_k p^T J - J^T p \nabla_x f_k^T \\
&\quad + J^T pp^T J) \\
&= \sum_k p_k \nabla_x f_k \nabla_x f_k^T - (\sum_k p_k \nabla_x f_k) p^T J \\
&\quad - J^T p(\sum_k p_k \nabla_x f_k)^T + J^T pp^T J \\
&= \sum_k p_k \nabla_x f_k \nabla_x f_k^T - J^T pp^T J - J^T pp^T J + J^T pp^T J \\
&= \sum_k p_k J_k^T J_k - J^T pp^T J \\
&= J^T \operatorname{diag}(p) J - J^T pp^T J \\
&= J^T(\operatorname{diag}(p) - pp^T)J \\
&= H(x)
\end{aligned}$$

This concludes the proof. □

Thus, minimizing with the Gaussian noise regularizer (9) is equivalent to minimizing the curvature in all directions equally since $\mathrm{Tr}(H(x)) = \sum_i \lambda_i(H(x))$ where λ_i denote the sorted Hessian eigenvalues. This has the effect of inducing low curvature in the loss landscape of $\mathcal{L}$ around x and encourages a locally linear behavior.

Substituting (8) into (9) and simplifying, we obtain:

$$\mathop{\mathbb{E}}_{\sigma\sim\mathcal{U}([0,\sigma_{\max}])} \mathop{\mathbb{E}}_{\delta\sim\mathcal{N}(0,\sigma^2)} D(p_\theta(x) \parallel p_\theta(x+\delta)) \approx \frac{\sigma^2_{\max}}{6} \sum_k [p_\theta(x)]_k \|\nabla_x \log[p_\theta(x)]_k\|_2^2 \quad (11)$$

This can be interpreted a regularization term that induces stability of predictions within a local neighborhood of x through weighted logit smoothing. This type of weighted logit smoothing leads to a bound on the local loss deviation.

Theorem 1 *The following bound holds on the loss function:*

$$|\mathcal{L}(x+\delta,y)-\mathcal{L}(x,y)| \leq \|\delta\|_2 \cdot \sqrt{\sum_k y_k \|\nabla_x \log p(x)_k\|_2^2} + \frac{1}{2}\lambda_{max}(H(x))\|\delta\|_2^2 + o(\|\delta\|_2^2) \quad (12)$$

Proof: Using the quadratic loss approximation near x, we have:

$$\mathcal{L}(x+\delta,y) = \mathcal{L}(x,y)+\delta^T\nabla_x\mathcal{L}(x,y)+\frac{1}{2}\delta^T H(x)\delta+o(\|\delta\|_2^2)$$

By using up to second order terms, we can obtain an upper bound on the loss variation as:

$$\begin{aligned} &|\mathcal{L}(x+\delta,y) - \mathcal{L}(x,y)| \\ &\approx |\langle \nabla_x\mathcal{L}(x,y),\delta\rangle + \frac{1}{2}\delta^T H(x)\delta| \\ &\leq \|\delta\|_2 \cdot \|\nabla_x\mathcal{L}(x,y)\|_2 + \frac{1}{2}\delta^T H(x)\delta \end{aligned} \quad (13)$$

where we used the Cauchy-Schwarz inequality to bound the linear term and the fact that H is positive semidefinite. The gradient in the first term of (13) can be written as:

$$\begin{aligned} &\|\nabla_x\mathcal{L}(x,y)\|_2 \\ &= \sqrt{\|\sum_k -y_k \nabla_x \log p(x)_k\|_2^2} \\ &= \sqrt{\sum_k y_k \|\nabla_x \log p(x)_k\|_2^2} \\ &= \sqrt{\sum_k e_k \|\nabla_x \log p(x)_k\|_2^2 + \sum_k p(x)_k \|\nabla_x \log p(x)_k\|_2^2} \end{aligned}$$

where $e_k = y_k - p(x)_k$ is the prediction error. Upper bounding the quadratic term in (13), we obtain:

$$\begin{aligned} \delta^T H(x)\delta &= \delta^T \left(\sum_i \lambda_i(H(x)) u_i u_i^T\right)\delta \\ &= \sum_i \lambda_i(H(x))(\delta^T u_i)^2 \\ &\leq \lambda_{\max}(H(x)) \sum_i (\delta^T u_i)^2 \\ &= \lambda_{\max}(H(x))\|\delta\|_2^2 \end{aligned}$$

Using the two preceding bounds with (13), we obtain the desired bound (12). The proof is complete. □

A consequence of Theorem 1 is that for correct classifications where $y_k \approx p(x)_k$, minimizing the regularizer (11) has the effect of (a) minimizing the curvature, as the Hessian trace upper bounds the maximum Hessian eigenvalue of the loss $\mathcal{L}(x,y)$, $\lambda_{\max}(H(x)) \leq \mathrm{Tr}(H(x))$, in addition to (b) increasing loss surface flatness by minimizing the norm of the loss gradient, $\|\nabla_x\mathcal{L}(x,y)\|_2$. Thus, we choose to include the effect of *true class confidence* in this regularizer as shown in the second term of (6) via a multiplication with $p(x)_y$. This adapts the regularization strength making this regularizer misclassification-aware. For higher confidence examples, the Gaussian smoothing effect increases to make the loss more locally regular, while for misclassifications this effect is minimized to focus the learning on the classification loss.

6.3. Experimental Results

Robustness Results. Figure 4 shows a comparison between clean and mean corrupted accuracies on CIFAR-10 for various deep learning methods using a ResNet-18 architecture. Figure 2 visualizes the corruption robustness profile; it shows that the combination of our MAGN regularizer with AugMix improves corruption robustness upon standard models by a large margin. Table 5 and Figures 5, 3 showcase additional results on the clean and corrupted CIFAR-100 datasets, similar to those shown earlier on CIFAR-10, for the same baselines on the ResNet-18 architecture. Compared to standard training, MAGN achieves an mCA absolute improvement of 8.2%, and when combined with AugMix, it increases to 15.9%. Again, this improves upon the previous SOTA AugMix which achieves a smaller 12.1% improvement, and adversarial training variants.

Uncertainty Calibration Results. Calibration performance on the clean and corrupted CIFAR-10 validation sets on different model architectures is shown in Figure 6.

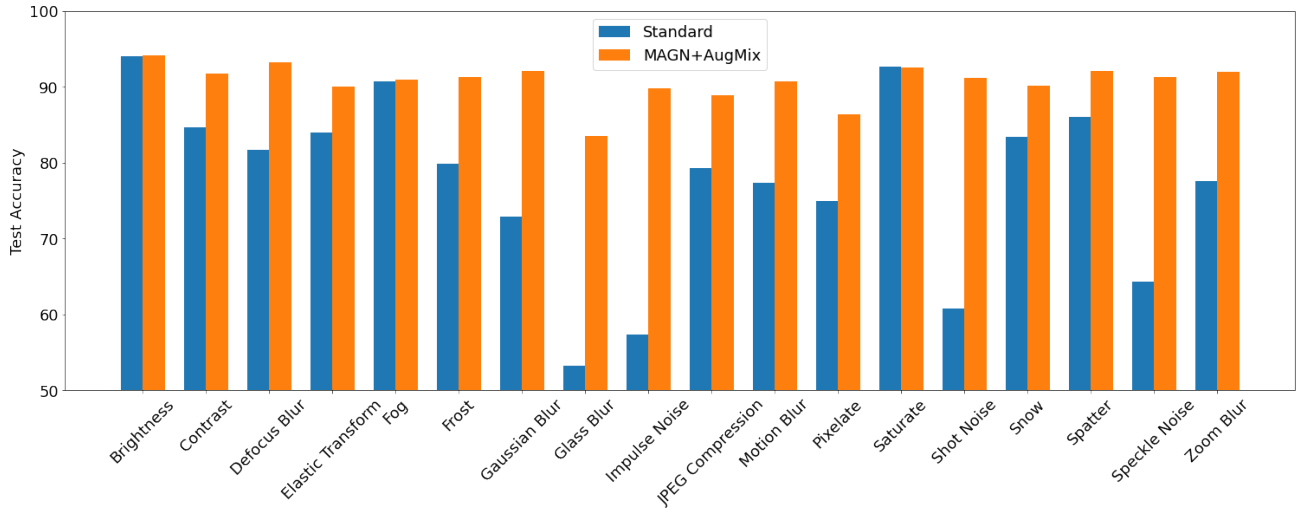

Figure 2. Robustness against a variety of corruptions including CIFAR-10-C weather, blur, noise and digital corruptions. The combination MAGN + AugMix improves robustness by a large margin despite not having seen these in the training process.

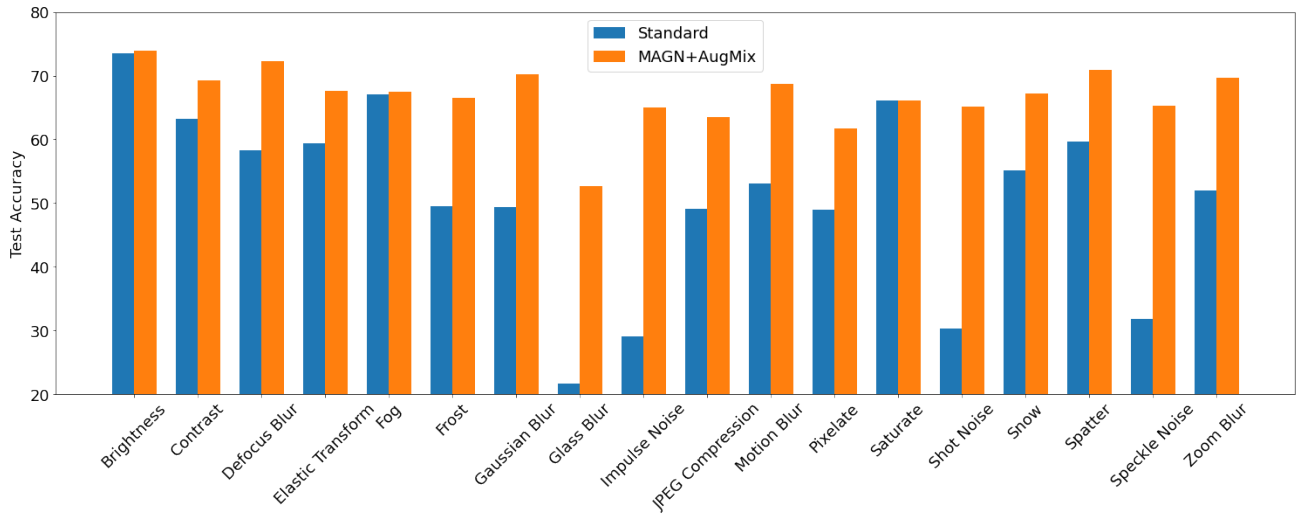

Figure 3. Robustness against a variety of corruptions including CIFAR-100-C weather, blur, noise and digital corruptions. Similar to the observed results on the CIFAR-10-C dataset, our combination MAGN + AugMix yields improved robustness.

Hyperparameter Sensitivity. We study the effect of the training hyperparameters $(\lambda_1, \lambda_2, \sigma_{\max})$ on the robustness and uncertainty estimation performance on the ResNet-18 architecture trained for the CIFAR-10 dataset where we use the training loss defined in (14):

$$\min_{\theta} \mathop{\mathbb{E}}_{(x,y)\sim\mathcal{D}} \Big[\mathcal{L}\left(f_\theta(x), y\right) + \lambda_1 [p_\theta(x)]_y \mathop{\mathbb{E}}_{\sigma\sim\mathcal{U}([0,\sigma_{\max}])} \mathop{\mathbb{E}}_{\delta\sim\mathcal{N}(0,\sigma^2)} D(p_\theta(x) \parallel p_\theta(x+\delta)) \Big] + \lambda_2 \cdot JS(p_\theta(x); p_\theta(x_{a,1}); p_\theta(x_{a,2})) \Big]. \tag{14}$$

The classification loss $\mathcal{L}$ in (14) is the standard term that maximizes accuracy on clean examples and the first regularizer is our proposed MAGN consistency loss. The second Jensen-Shannon regularizer embeds the original and augmented examples similarly, and the augmentations are generated using the mixing chains of AugMix (Hendrycks et al., 2020).

Table 4 contains the results from which it is evident that the robustness and calibration is not too sensitive to the choice of hyperparameters as the clean accuracy, mCA, and RMSE calibration error do not change significantly.

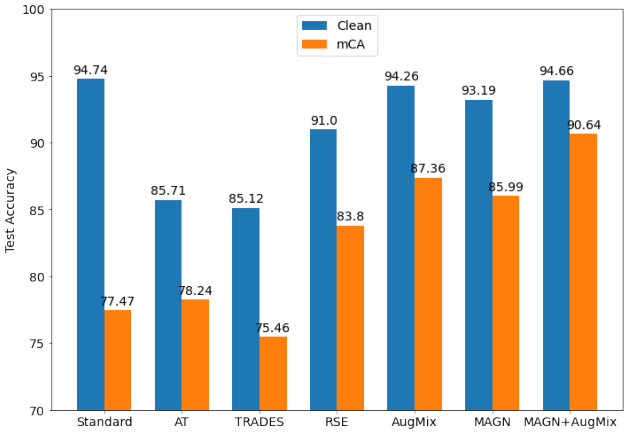

Figure 4. Clean and mCA performance on CIFAR-10 and CIFAR-10-C respectively for various deep learning methods. Our MAGN approach provides a significant robustness improvement in comparison to the standard model. A further combination of MAGN+AugMix yields state of the art performance.

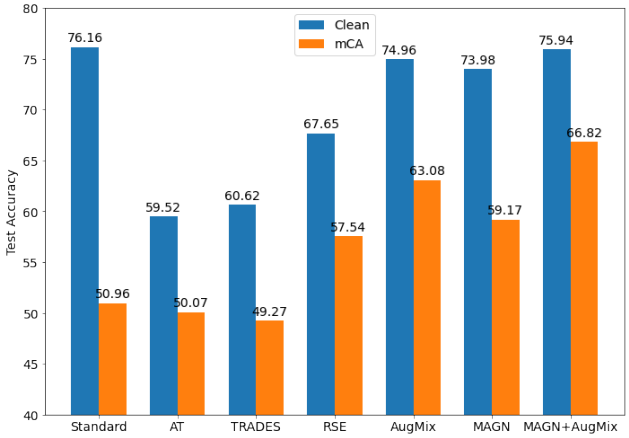

Figure 5. Clean and mCA performance on CIFAR-100 and CIFAR-100-C respectively for various deep learning methods. Similar to Figure 4 for CIFAR-10-C, our approach MAGN provides robustness improvement over the standard model and the combination MAGN+AugMix yields additional gains in performance.

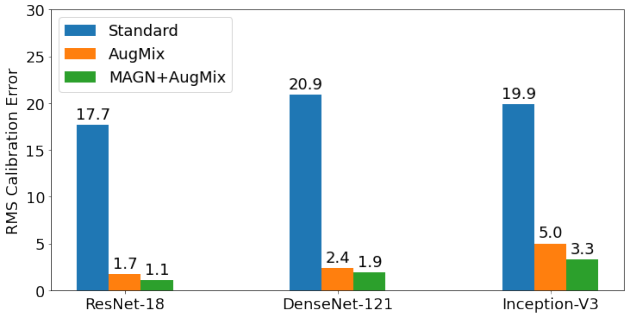

Figure 6. RMS calibration error performance of various deep learning architectures on the corrupted dataset CIFAR-10-C. The combination MAGN+AugMix significantly improves the uncertainty calibration on unforeseen corruptions in comparison to standard models.

Parameters	Clean Acc.	mCA	Clean RMS Cal. Error	Corrupt RMS Cal. Error
(3, 12, 0.2)	94.66	90.64	2.2	1.1
(3, 12, 0.4)	94.79	90.74	2.7	1.1
(5, 12, 0.2)	94.69	90.70	2.3	0.9
(3, 10, 0.2)	94.48	90.61	1.4	2.1

Table 4. Sensitivity analysis on hyperparameters of MAGN + AugMix. Test set accuracy and RMS calibration error for CIFAR-10 and CIFAR-10-C datasets on ResNet-18 architecture.

Corruption	Standard	AT	TRADES	RSE	AugMix	MAGN	MAGN+AugMix
Clean	**76.16**	59.52	60.62	67.65	74.96	73.98	75.94
Brightness	73.53	56.24	57.27	65.13	72.71	70.93	**73.90**
Contrast	63.22	29.08	23.69	39.83	66.97	53.04	**69.26**
Defocus Blur	58.32	54.04	52.55	59.84	71.23	56.98	**72.28**
Elastic Transform	59.38	52.39	50.73	57.92	66.12	58.37	**67.55**
Fog	67.05	37.27	32.20	45.45	65.87	60.89	**67.42**
Frost	49.55	50.79	49.49	60.69	60.56	64.86	**66.48**
Gaussian Blur	49.37	52.14	49.79	56.86	68.86	50.41	**70.25**
Glass Blur	21.66	52.23	50.41	**54.24**	40.47	47.70	52.67
Impulse Noise	29.05	39.63	43.04	53.47	56.04	59.94	**65.06**
JPEG Compression	49.09	57.02	56.83	**64.13**	61.83	59.80	63.43
Motion Blur	53.11	50.36	47.72	54.58	66.85	50.49	**68.71**
Pixelate	49.01	57.37	57.36	**64.46**	58.87	59.47	61.70
Saturate	**66.12**	48.88	50.84	56.63	64.99	63.93	66.06
Shot Noise	30.28	53.00	54.42	64.23	55.15	**66.32**	65.14
Snow	55.11	53.60	53.03	59.27	64.76	61.39	**67.14**
Spatter	59.67	52.94	53.77	57.53	69.64	62.48	**70.95**
Speckle Noise	31.78	51.58	52.97	63.40	56.21	**66.09**	65.24
Zoom Blur	51.94	52.75	50.76	58.20	68.37	51.92	**69.62**
mCA	50.96	50.07	49.27	57.54	63.08	59.17	**66.82**

Table 5. Classification accuracy on CIFAR-100-C across different corruptions for various methods ResNet18 architecture. The best accuracy is highlighted in bold for each row which corresponds to a specific corruption. MAGN + AugMix achieves the highest mCA score.