CEB IMPROVES MODEL ROBUSTNESS

Anonymous authors
Paper under double-blind review

ABSTRACT

We demonstrate that the Conditional Entropy Bottleneck (CEB) can improve model robustness. CEB is an easy strategy to implement and works in tandem with data augmentation procedures. We report results of a large scale adversarial robustness study on CIFAR-10, as well as the IMAGENET-C Common Corruptions Benchmark.

1 INTRODUCTION

We aim to make models that make meaningful predictions beyond the data they were trained on. Generally we want our models to be *robust*. Broadly, robustness is the ability of a model to continue making valid predictions as the distribution the model is tested on moves away from the empirical training set distribution. The most commonly reported robustness metric is simply test set performance, where we verify that our model continues to make valid predictions on what we hope represents valid draws from the exact same data generating procedure.

Adversarial robustness tests robustness in a worst case setting, where an attacker (Szegedy et al., 2013) makes limited targeted modifications to the input that are are fooling as possible. Many adversarial attacks have been proposed and studied (Szegedy et al., 2013; Carlini & Wagner, 2017b;a; Kurakin et al., 2016a; Madry et al., 2017). Essentially all machine-learned systems are currently believed by default to be highly vulnerable to adversarial examples. Many defenses have been proposed, but very few have demonstrated robustness against a powerful, general-purpose adversary (Carlini & Wagner, 2017a; Athalye et al., 2018). While robustness to adversarial attacks continues to attract interest, recent discussions have emphasized the need to consider other forms of robustness as well (Engstrom et al., 2019). The Common Corruptions Benchmark (Hendrycks & Dietterich, 2019) measures image models robustness to more mild but real world sorts of perturbations. Even these modest perturbations can be very fooling for traditional architectures.

One of the few general purpose strategies that demonstrably improves model robustness is Data Augmentation (Cubuk et al., 2018; Lopes et al., 2019; Yin et al., 2019). However, it would be nice to identify loss-based solutions that can work in tandem with the data augmentation approaches. Intuitively by performing modifications of the inputs at training time, the model is prevented from being too sensitive to particular features of the inputs that don't survive the augmentation procedure.

Alternatively, we can try to make our models more robust by making them less sensitive to the inputs in the first place. The goal of this work is to experimentally investigate whether by systematically limiting the complexity of the extracted representation we can make our models more robust in all three of these senses: test set generalization, worst-case robustness, and typical-case robustness.

1.1 CONTRIBUTIONS

This paper is primarily empirical. We demonstrate:

- CEB models are easy to implement and train.
- CEB models demonstrate improved generalization performance over deterministic baselines on CIFAR-10 and Imagenet.
- CEB models show improved robustness to adversarial attacks on CIFAR-10.
- CEB models show improved robustness on the IMAGENET-C Common Corruptions Benchmark.

2 BACKGROUND

2.1 INFORMATION BOTTLENECKS

The information bottleneck objective (Tishby et al., 2000) aims to learn a stochastic representation $Z \sim p(z|x)$ that retains as much information about a target variable Y while being as compressed as possible. The objective:[1]

$$\max I(Z;Y) - \sigma(-\rho)I(Z;X), \tag{1}$$

uses a Lagrange multiplier $\sigma(-\rho)$ to trade off between the relevant information $(I(Z;Y))$ and complexity of the representation $(I(Z;X))$. Because Z depends only on X ($Z \leftarrow X \leftrightarrow Y$): Z and Y are conditionally independent given Z:

$$I(Z;X,Y) = I(Z;X) + \cancel{I(Z;Y|X)} = I(Z;Y) + I(Z;X|Y). \tag{2}$$

This allows us to write the information bottleneck of Equation (1) in an equivalent form:

$$\max I(Z;Y) - e^{-\rho}I(Z;X|Y). \tag{3}$$

Just as the original Information Bottleneck objective (Equation (1)) admits a natural variational lower bound (Alemi et al., 2017), so does this form. We can variationally lower bound the mutual information between our representation and the targets with a variational decoder $q(y|z)$:

$$I(Z;Y) = \mathbb{E}_{p(x,y)p(z|x)}\left[\log\frac{p(y|z)}{p(y)}\right] \geq H(Y) + \mathbb{E}_{p(x,y)p(z|x)}\left[\log q(y|z)\right]. \tag{4}$$

While we may not know $H(Y)$ exactly for real world datasets, in the information bottleneck formulation it is a constant outside of our control and so can be dropped in our objective. We can variationally upper bound our residual information:

$$I(Z;X|Y) = \mathbb{E}_{p(x,y)p(z|x)}\left[\log\frac{p(z|x,y)}{p(z|y)}\right] \leq \mathbb{E}_{p(x,y)p(z|x)}\left[\log\frac{p(z|x)}{q(z|y)}\right], \tag{5}$$

with a variational class conditional marginal $q(z|y)$ that approximates $\int dx\, p(z|x)p(x|y)$. Putting both bounds together gives us the Conditional Entropy Bottleneck Objective (Fischer, 2018):

$$\min_{p(z|x)} \mathbb{E}_{p(x,y)p(z|x)}\left[\log q(y|z) - e^{-\rho}\log\frac{p(z|x)}{q(z|y)}\right] \tag{6}$$

Compare this with the VIB Objective (Alemi et al., 2017):

$$\min_{p(z|x)} \mathbb{E}_{p(x,y)p(z|x)}\left[\log q(y|z) - \sigma(-\rho)\log\frac{p(z|x)}{q(z)}\right]. \tag{7}$$

The difference between CEB and VIB is the presence of a class conditional versus unconditional variational marginal. As can be seen in Equation (5): using an unconditional marginal provides a looser variational upper bound on $I(Z;X|Y)$. CEB (Equation (6)) can be thought of as a tighter variational approximation than VIB (Equation (7)) to Equation (3). Since Equation (3) is equivalent to the IB objective (Equation (1)), CEB can be thought of as a tighter variational approximation to the IB objective than VIB.

2.2 IMPLEMENTING A CEB MODEL

In practice, turning an existing classifier architecture into a CEB model is very simple. For the stochastic representation $p(z|x)$ we simply use the original architecture, replacing the final softmax layer with a dense layer with d outputs. These outputs are then used to specify the means of a d-dimensional Gaussian distribution with unit diagonal covariance. That is, to form the stochastic representation, independent standard normal noise is simply added to the output of the network ($z = x + \epsilon$). For every input, this stochastic encoder will generate a random d-dimensional output

[1] The IB objective is ordinarily written with a Lagrange multiplier $\beta \equiv \sigma(-\rho)$ with a natural range from 0 to 1. Here we use the sigmoid function: $\sigma(-\rho) \equiv \frac{1}{1+e^{\rho}}$ to reparameterize in terms of a control parameter ρ on the whole real line. As $\rho \to \infty$ the Bottleneck turns off.

vector. For the variational classifier $q(y|z)$ any classifier network can be used, including just a linear softmax classifier as done in these experiments. For the variational conditional marginal $q(z|y)$ it helps to use the same distribution as output by the classifier. For the simple unit variance Gaussian encoding we used in these experiments, this requires learning just d parameters per class. For ease of implementation, this can be represented as single dense linear layer mapping from a one-hot representation of the labels to the d-dimensional output, interpreted as the mean of the corresponding class marginal.

In this setup the CEB loss takes a particularly simple form:

$$\mathbb{E}\left[w_y \cdot (f(x)+\epsilon) - \log \sum_{y'} e^{w_{y'} \cdot (f(x)+\epsilon)} - \frac{e^{-\rho}}{2}(f(x)-\mu_y)(f(x)-\mu_y+2\epsilon)\right]. \quad (8)$$

Here the first term is the usual softmax classifier loss, but acting on our stochastic representation $z = f(x) + \epsilon$, which is simply the output of our encoder network $f(x)$ with additive Gaussian noise. The w_y is the yth row of weights in the final linear layer outputing the logits. μ_y are the learned class conditional means for our marginal. ϵ are standard normal draws from an isotropic unit variance Gaussian with the same dimension as our encoding $f(x)$. The second term in the loss is a stochastic sampling of the KL divergence between our encoder likelihood and the class conditional marginal likelihood. ρ controls the strength of the Bottleneck and can vary on the whole real line. As $\rho \to \infty$ the Bottleneck is turned off. In practice we find that ρ values near but above 0 tend to work best for modest size models, with the tendency for the best ρ to approach 0 as the model capacity increases. Notice that in expectation the second term in the loss is $(f(x) - \mu_y)^2$, which encourages the learned means μ_y to converge to the average of the representations of each element in the class. During testing we simply used the mean encodings and removed the stochasticity.

In its simplest form, CEB training a classifier amounts to injecting Gaussian random noise in the penultimate layer and learning estimates of the class averaged output of that layer with the stochastic regularization shown.

2.3 Adversarial Attacks and Defenses

Attacks. The first adversarial attacks were proposed in Szegedy et al. (2013); Goodfellow et al. (2015). Since those seminal works, an enormous variety of attacks has been proposed (Kurakin et al. (2016a;b); Moosavi-Dezfooli et al. (2016); Carlini & Wagner (2017b); Madry et al. (2017); Eykholt et al. (2017); Baluja & Fischer (2017), etc.). In this work, we will primarily consider the Projected Gradient Descent (PGD) attack (Madry et al., 2017), which is a multi-step variant of the early Fast Gradient Method (Goodfellow et al., 2015). The attack can be viewed as having four parameters: p, the norm of the attack (typically 2 or ∞), ϵ, the radius the the p-norm ball within which the attack is permitted to make changes to an input, n, the number of gradient steps the adversary is permitted to take, and ϵ_i, the per-step limit to modifications of the current input. In this work, we consider L_2 and L_∞ attacks of varying ϵ and n, and with $\epsilon_i = \frac{4}{3}\frac{\epsilon}{n}$.

Defenses. A common defense for adversarial examples is Adversarial training. Adversarial training was originally proposed in Szegedy et al. (2013), but was not practical until the Fast Gradient Method was introduced. It has been studied in detail, with varied techniques (Kurakin et al., 2016b; Madry et al., 2017; Ilyas et al., 2019; Xie et al., 2019). Adversarial training can clearly be viewed as a form of data augmentation (Tsipras et al., 2018), where instead of using some fixed set of functions to modify the training examples, we use the model itself in combination with one or more adversarial attacks to modify the training examples. As the model changes, the distribution of modifications changes as well. However, unlike with non-adversarial data augmentation techniques, such as AutoAug, adversarial training techniques considered in the literature so far cause substantial reductions in accuracy on clean test sets. For example, the CIFAR10 model described in Madry et al. (2017) gets 95.5% accuracy when trained normally, but only 87.3% when trained on L_∞ adversarial examples. More recently, Xie et al. (2019) adversarially trains ImageNet models with impressive robustness to targeted PGD L_∞ attacks, but at only 62.32% accuracy on the non-adversarial test set, compared to 78.81% accuracy for the same model trained only on clean images.

2.4 Common Corruptions

The Common Corruptions Benchmark (Hendrycks & Dietterich, 2019) offers a real world test of model robustness in light of common image processing pipeline corruptions. Figure 4 shows examples of the 15 corruptions present in the benchmark. ImageNet-C is a modified test set of Imagenet images with the 15 corruptions applied at five different strengths. Within each corruption type we evaluated the average error at each of the five levels ($E_c = \frac{1}{5}\sum_{s=1}^{5} E_{cs}$). To summarize the performance across all corruptions we report not only the average corruption error across all 15 tasks (avg $= \frac{1}{15}\sum_c E_c$), but also the commonly reported *Mean Corruption Error* (mCE), which reweights the errors on each task according to the performance of a baseline AlexNet model (Hendrycks & Dietterich, 2019):

$$\text{mCE} = \frac{1}{15}\sum_c \frac{\sum_{s=1}^{5} E_{cs}}{\sum_{s=1}^{5} E_{cs}^{\text{AlexNet}}}. \tag{9}$$

There are slightly different pipelines that have been used in the literature for the ImageNet-C task (Lopes et al., 2019). In this work we used the AlexNet normalization numbers and data formulation as in Yin et al. (2019).

3 Experiments

3.1 Fashion-MNIST Experiments

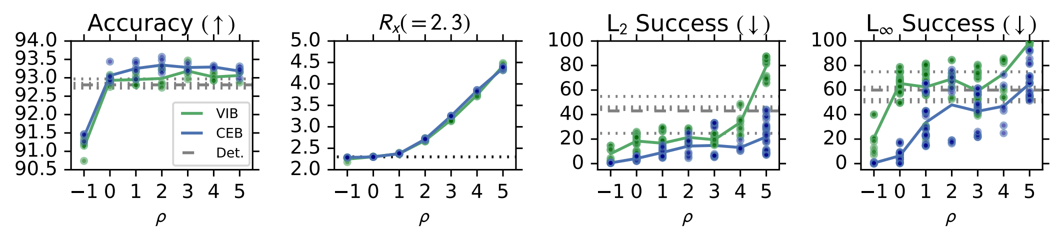

Figure 1: Clean accuracy, max training rate $R_X > I(Z;X)$, and robustness to targeted PGD L_2 and L_∞ attacks on CEB, VIB, and Deterministic models trained on Fashion MNIST. We can see that at any given value of ρ, the CEB models dominate the VIB models on both accuracy and robustness, while having essentially identical maximum rates. None of these models are adversarially trained.

As a warm up, we consider Wide ResNet (Zagoruyko & Komodakis, 2016) models trained on Fashion MNIST (Xiao et al., 2017), and evaluated on targeted PGD L_2 and L_∞ attacks. All of the attacks are targeting the *trouser* class of Fashion MNIST, as that is the most distinctive class. Targeting a less distinctive class, such as one of the shirt classes, would confuse the difficulty of classifying the different shirts and the robustness of the model to adversaries. To measure robustness to the targeted attacks, we count the number of predictions that changed from a correct prediction on the clean image to an incorrect prediction of the target class on the adversarial image, and divide by the original number of correct predictions. Results are shown in Figure 1.

In this experiment we wanted to compare the performance of VIB, CEB, and a deterministic baseline. In Figure 1 (left) we see that both VIB and CEB have improved accuracy over the deterministic baseline. In order to compare the relative complexity of the learned representations for the two models, in the second panel we demonstrate the *rate*: $\mathbb{E}\left[\log \frac{p(z|x)}{q(z)}\right] > I(Z;X)$ using a trained variational marginal for both VIB and CEB. Since CEB does not include a marginal in the objective, we train the marginal with a separate optimizer to prevent it from influencing the CEB model's parameters. The rate is a variational upper bound on the mutual information between the input and the stochastic representation. The two models show nearly the same rate.

The right two panels of Figure 1 show robustness to the targeted PGD L_2 and L_∞ attacks. Here CEB outperforms VIB. We also see for both models that as ρ decreases, the robustness to both attacks increases. In line with the proposed Minimum Necessary Information criterion from Fischer (2018), at $\rho = 0$ we end up with CEB models that have hit exactly 2.3 nats for the rate, have maintained high accuracy, and have strong robustness to both attacks. Moving to $\rho = -1$ gives only a small improvement to robustness, at the cost of a large decrease in accuracy.

3.2 CIFAR10 EXPERIMENTS

28x10 Wide ResNet Experiments We trained a set of 25 28x10 Wide ResNet (WRN) CEB models on CIFAR10 at $\rho \in [-1, -0.75, ..., 5]$, as well as a deterministic baseline. They trained for 1500 epochs, lowering the learning rate by a factor of 0.3 after 500, 1000, and 1250 epochs. This long training regime was due to our use of the original AUTOAUG policies, which requires longer training. The only additional modification we made to the basic 28x10 WRN architecture was the removal of all Batch Normalization (Ioffe & Szegedy, 2015) layers. Every small model we have trained with Batch Normalization enabled has had substantially worse robustness, even though typically the accuracy is much higher. For example, 28x10 WRN CEB models rarely exceeded more than 10% adversarial accuracy. However, it was always still the case that lower values of ρ gave higher robustness. Additional training details are in Appendix A.

Figure 2 demonstrates the adversarial robustness of CEB models to both targeted L_2 and L_∞ attacks. The CEB models show a marked improvement in robustness to L_2 attacks compared to an adversarially-trained baseline from Madry et al. (2017) (denoted Madry). Figure 3 shows the robustness of five of those models to PGD attacks as ϵ is varied. We selected the four CEB models to represent the most robust models across most of the range of ρ we trained. Note that of the 25 CEB models we trained, only the models with $\rho \geq 1$ succesfully trained. The remainder collapsed to chance performance. This is something we observe on all datasets when training models that are too low capacity. Only by increasing model capacity does it become possible to train at low ρ. Note that this result is predicted by the theory of the onset of learning in the Information Bottleneck and its relationship to model capacity from Wu et al. (2019).

62x7 Wide ResNet Experiments. In order to explore the effect of model size on training, and to train at lower ρ, we trained the largest Wide ResNet we could fit on a single GPU with a batch size of 250. This was a 62×7 model similar to the ones above, including the use of AUTOAUG, but we additionally enabled batch normalization. We were able to train at $\rho = 0$ with this larger model, which reached **97.51%** accuracy. This result is better than the 28x10 Wide ResNet from AUTOAUG by 0.19 percentage points, although it is still worse than the Shake-Drop model from that paper. We additionally tested the model on the new CIFAR10.1 test set (Recht et al., 2018), getting accuracy of 93.6%. This is a gap of **3.9** percentage points, which is better than all of the results reported in that paper, and substantially better than the Wide ResNet results (but still inferior to the Shake-Drop AUTOAUG results). The same model at $\rho = 5$ reached 97.05% accuracy on the normal test set and 91.9% on the CIFAR10.1 test set, showing that increased ρ gave substantially worse generalization.

To test robustness of these models, we swept ϵ for both PGD attacks, which we show in Figure 3. The main result is that the 62×7 CEB_0 model not only has substantially higher accuracy than baseline Wide ResNets trained with AUTOAUG, it also beats the adversarially-trained model on both the L_2 and the L_∞ attacks at almost all values of ϵ. We also show that this model is even more robust to two transfer attacks, where we used the 62×7 CEB_5 model and the adversarially-trained model to generate PGD attacks, and then test them on the CEB_0 model. This result helps to counter possible claims that these models are doing “gradient masking” (the more compelling evidence against gradient masking is that the robustness of the model is strongly correlated with the hyperparameter ρ, whose only effect is to constrain the amount of information the model captures).

We additionally tested both models on the CIFAR10 Common Corruptions test sets. At the time of training, we were unaware that AUTOAUG’s default policies for CIFAR10 contains brightness and contrast augmentations that amount to training on those two corruptions from Common Corruptions (as mentioned in Yin et al. (2019)), so our results are not appropriate for direct comparison with other results in the literature. However, they still allow us to compare the effect of bottlenecking the information between these two large models. The $\rho = 5$ model reached an mCE[2] of 61.2. The $\rho = 0$ model reached an mCE of 52.0, which is a dramatic relative improvement.

[2]The mCE is the mean Corruption Error across the 15 classes of corruption relative to a baseline model. We use the baseline model presented in Yin et al. (2019).

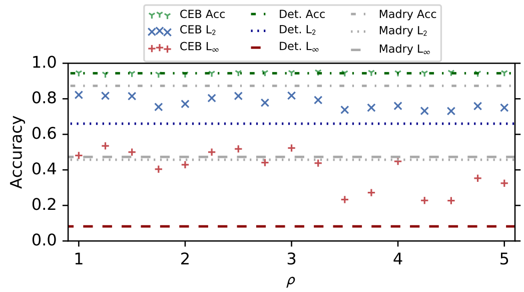

Figure 2: CEB ρ vs. test set accuracy, and L_2 and L_∞ PGD adversarial attacks on CIFAR-10. The attack parameters were selected to be about equally difficult for the adversarially-trained WRN 28x10 model from Madry et al. (2017) (grey dashed and dotted lines). The deterministic baseline (Det.) only gets 8% accuracy on the L_∞ attacks, but gets 66% on the L_2 attack, substantially better than the 45.7% of the adversarially-trained model, which makes it clear that the adversarially-trained model failed to generalize in any reasonable way to the L_2 attack. The CEB models are always substantially more robust than Det., and many of them outperform Madry even on the L_∞ attack the Madry model was trained on, but for both attacks there is a clear general trend toward more robustness as ρ decreases. Finally, the CEB and Det. models all reach about the same accuracy, ranging from 93.9% to 95.1%, with Det. at 94.4%. In comparison, Madry only gets 87.3%.

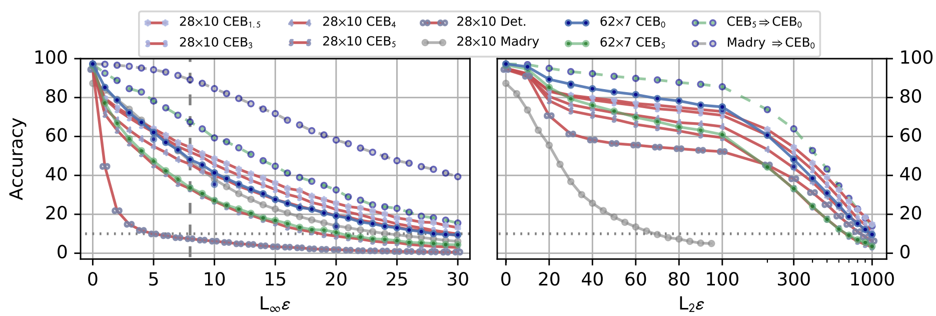

Figure 3: Untargeted adversarial attacks on small and large CIFAR10 models showing both strong robustness to PGD L_2 and L_∞ attacks, as well as excellent test accuracy of 97.5%. **Left:** Accuracy on untargeted L_∞ attacks at different values of ε for all 10,000 test set examples. 28×10 and 62×7 indicate the Wide ResNet size. CEB_x indicates a CEB model trained at $\rho = x$. Madry is the adversarially-trained model from Madry et al. (2017) (values provided by Aleksander Madry). Madry was trained with 7 steps of L_∞ PGD at $\varepsilon = 8$ (grey dashed line). CEB_0 is the CEB model with the highest accuracy (97.51%) trained at $\rho = 0$. CEB_5 is the CEB model with the highest accuracy (97.05%) trained at $\rho = 5$. $\text{CEB}_5 \Rightarrow \text{CEB}_0$ is transfer attacks from the 62×7 CEB_5 model to the 62×7 CEB_0 model. Madry $\Rightarrow \text{CEB}_0$ is transfer attacks from the Madry model to the 62×7 CEB_0 model. All of the CEB models with $\rho \leq 4$ outperform Madry across most of the values of ϵ, even though they were not adversarially-trained. The 62×7 CEB_0 model is even more robust to transfer attacks than to the direct whitebox attacks. **Right:** Accuracy on untargeted L_2 attacks at different values of ε. Note the switch to log scale on the x axis at $L_2\epsilon = 100$. The extra values at $L_\infty\varepsilon \in [5, 10, 20]$ and $L_2\varepsilon \in [30, 100, 300]$ for CEB_0 are collected from 50 steps of PGD, showing that extra gradient steps provide minimal benefit to the attacker against this model. All other values are collected at 20 steps of PGD. It is interesting to note that both the small and the large CEB_5 have essentially identical robustness, in spite of very different numbers of parameters, and that the Det. model eventually outperforms the CEB_5 models on L_2 attacks at relatively high accuracies. We emphasize that none of our models are adversarially trained.

3.3 IMAGENET EXPERIMENTS

To demonstrate CEB's ability to improve robustness to real world data shifts, we utilized the IMAGENET-C Bechmark task. We ran six different types of networks. As a simple baseline we ran RESNET-50 style networks with no data augmentation. We then ran the same networks but as CEB networks at ten different values of $\rho = (1, 2, \ldots, 10)$. AUTOAUG (Cubuk et al., 2018) has previously been demonstrated to improve robustness markedly on IMAGENET-C so next we retrained our baseline RESNET-50 model but with AUTOAUG. We similarly trained these AUTOAUG models as CEB models with ten different values of ρ as before. Finally, IMAGENET-C numbers are also sensitive to the model capacity. To asses whether CEB can benefit larger models we next repeated the experiments with a modified RESNET-50 network where every layer was made twice as wide. All of the results are summarized in Figure 4 and Table 1.

type	CEBx2	50x2	CEB	50	CEB-aa	50-aa
ρ^*	3	NA	7	NA	4	NA
Clean	**19.6**	21.9	**21.2**	22.1	**22.2**	23.2
mCE	**60.1**	64.7	**64.6**	66.4	**73.8**	78.8
Average CE	**47.5**	51.1	**51.0**	52.5	**58.5**	62.4
Gaussian Noise	**45.9**	49.8	**50.6**	52.0	**61.8**	70.2
Shot Noise	**45.7**	49.8	**50.1**	51.7	**64.1**	71.6
Impulse Noise	**48.3**	52.6	**52.3**	54.5	**66.6**	75.1
Defocus Blur	**57.3**	60.1	**60.0**	61.2	65.6	**65.3**
Glass Blur	**63.7**	67.0	**67.5**	67.8	**76.3**	77.2
Motion Blur	**53.9**	58.7	**57.7**	60.7	**61.6**	64.4
Zoom Blur	**60.3**	64.1	**63.4**	66.0	**62.1**	65.0
Snow	**58.0**	63.5	**62.9**	64.9	**69.7**	73.0
Frost	**52.7**	56.8	**57.4**	57.8	**61.8**	64.5
Fog	**34.9**	39.0	**37.5**	38.7	**45.8**	49.3
Brightness	**26.0**	28.9	**28.5**	29.1	**31.5**	33.2
Contrast	**48.3**	51.0	**51.7**	53.3	**61.7**	66.0
Elastic Transform	**41.7**	45.4	**44.9**	46.4	**48.3**	50.0
Pixelate	**38.2**	40.4	**40.7**	42.0	**56.0**	61.2
JPEG Compression	**36.9**	39.6	**39.9**	41.3	**44.6**	50.2

Table 1: This table reports values for each of the baseline and CEB models for the IMAGENET-C experiments. The middle columns show a RESNET-50 model trained with AUTOAUG ("50") versus the corresponding CEB network ("CEB"). The right columns ("-aa") remove AUTOAUG from the data processing pipeline, and the left columns ("x2") are models that are twice as wide. The CEB values reported here are the same ones denoted with the dots in Figure 4.

Both data augmentation and increasing model capacity have large positive effects on IMAGENET-C robustness, but for all three classes of models CEB gives substantial additional improvements.

4 CONCLUSION

The Conditional Entropy Bottleneck (CEB) provides a simple mechanism to improve robustness of image classifiers. We have shown that CEB gives a tighter variational bound on the IB objective than the closely-related VIB, while also having consistently better test accuracy and robustness. We have shown a strong trend toward increased robustness as ρ decreases in the standard 28×10 Wide ResNet model on CIFAR10, and that this increased robustness does not come at the expense of accuracy relative to the deterministic baseline. We have shown that CEB models at a range of ρ essentially dominate an adversarially-trained baseline model, even on the attack the adversarial model was trained on, and have incidentally shown that the adversarially-trained model generalizes to at least one other attack *less well* than a deterministic baseline. Finally, we have shown that on ImageNet, CEB provides substantial gains over deterministic baselines in both validation accuracy and robustness to Common Corruptions. We hope these empirical demonstrations inspire further theoretical and practical study of the use of bottlenecking techniques to encourage improvements to both classical generalization and robustness.

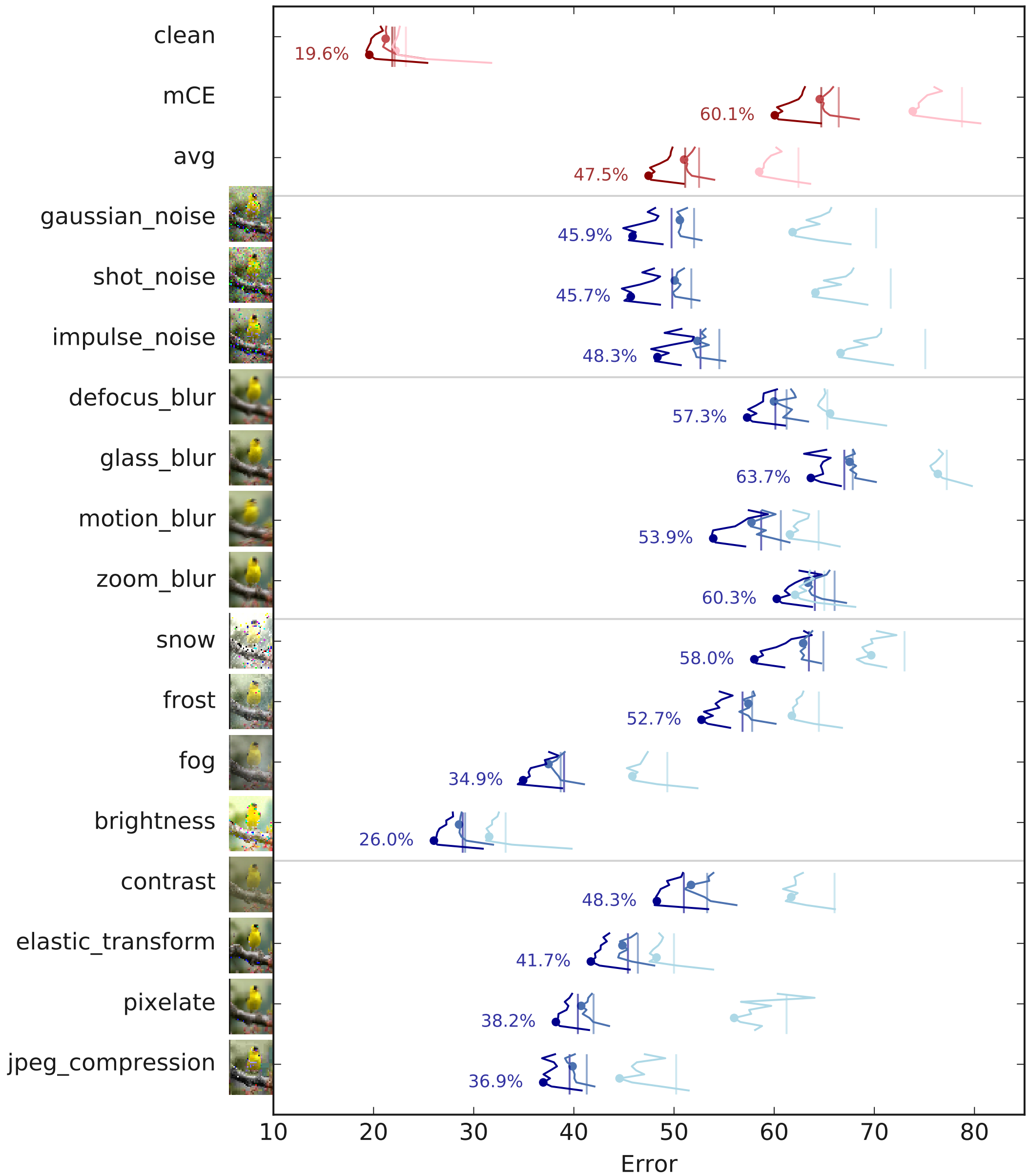

Figure 4: Summary of the IMAGENET-C experiments. In the main part of the image (in blue), the average errors (lower is better) across magnitude is shown for six different networks for each of the labeled Common Corruptions (Hendrycks & Dietterich, 2019). The networks come in pairs, with the vertical lines denoting the baseline network's performance and then in the corresponding color the errors for each of 10 different CEB trained networks are shown with varying $\rho = [1, 2, \ldots, 10]$, arranged from high to low. The light blue lines denote a baseline RESNET-50 model trained without AUTOAUG. The blue lines show the same network (and baseline) but with AUTOAUG. The dark blue lines show RESNET-50 networks that were made twice as wide. In this way we can separately see the effect of both data augmentation and enlarging the model, as well as the additive effect of CEB on each model. At the top in red are shown the same data for three summary statistics. `clean` denotes the clean errors of each of the networks. `mCE` denotes the ALEXNET regularized average corruption errors and `avg` shows a equally weighted average across all types of corruption. The dots denote the value for each CEB network and each corruption at ρ^*, the optimum ρ for the network as measured in terms of mCE. The values at these dots as well as the baseline values are given in detail in Table 1.

REFERENCES

Alexander A Alemi, Ian Fischer, Joshua V Dillon, and Kevin Murphy. Deep Variational Information Bottleneck. In *International Conference on Learning Representations*, 2017. URL http://arxiv.org/abs/1612.00410.

Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. *arXiv preprint arXiv:1802.00420*, 2018.

Shumeet Baluja and Ian Fischer. Adversarial transformation networks: Learning to generate adversarial examples. *arXiv preprint arXiv:1703.09387*, 2017.

Nicholas Carlini and David Wagner. Adversarial examples are not easily detected: Bypassing ten detection methods. In *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*, pp. 3–14. ACM, 2017a.

Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *2017 IEEE Symposium on Security and Privacy (SP)*, pp. 39–57. IEEE, 2017b.

Ekin D Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V Le. Autoaugment: Learning augmentation policies from data. *arXiv preprint arXiv:1805.09501*, 2018.

Logan Engstrom, Justin Gilmer, Gabriel Goh, Dan Hendrycks, Andrew Ilyas, Aleksander Madry, Reiichiro Nakano, Preetum Nakkiran, Shibani Santurkar, Brandon Tran, Dimitris Tsipras, and Eric Wallace. A discussion of 'adversarial examples are not bugs, they are features'. *Distill*, 2019. doi: 10.23915/distill.00019. https://distill.pub/2019/advex-bugs-discussion.

Kevin Eykholt, Ivan Evtimov, Earlence Fernandes, Bo Li, Amir Rahmati, Chaowei Xiao, Atul Prakash, Tadayoshi Kohno, and Dawn Song. Robust physical-world attacks on deep learning models. *arXiv preprint arXiv:1707.08945*, 2017.

Ian Fischer. The Conditional Entropy Bottleneck. *Open Review*, 2018.

Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *CoRR*, 2015.

Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *arXiv preprint arXiv:1903.12261*, 2019.

Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry. Adversarial examples are not bugs, they are features. *arXiv preprint arXiv:1905.02175*, 2019.

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015. URL https://arxiv.org/abs/1412.6980.

Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial examples in the physical world. *arXiv preprint arXiv:1607.02533*, 2016a.

Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial machine learning at scale. *arXiv preprint arXiv:1611.01236*, 2016b.

Raphael Gontijo Lopes, Dong Yin, Ben Poole, Justin Gilmer, and Ekin D. Cubuk. Improving robustness without sacrificing accuracy with patch gaussian augmentation, 2019.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.

Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: a simple and accurate method to fool deep neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2574–2582, 2016.

Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do cifar-10 classifiers generalize to cifar-10? *arXiv preprint arXiv:1806.00451*, 2018.

C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus. Intriguing properties of neural networks. *arXiv: 1312.6199*, 2013. URL https://arxiv.org/abs/1312.6199.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

Naftali Tishby, Fernando C Pereira, and William Bialek. The information bottleneck method. *arXiv preprint physics/0004057*, 2000.

Dimitris Tsipras, Shibani Santurkar, Logan Engstrom, Alexander Turner, and Aleksander Madry. Robustness may be at odds with accuracy. *arXiv preprint arXiv:1805.12152*, 2018.

Tailin Wu, Ian Fischer, Isaac Chuang, and Max Tegmark. Learnability for the information bottleneck. *Uncertainty in AI*, 2019.

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv: 1708.07747*, 2017. URL https://arxiv.org/abs/1708.07747.

Cihang Xie, Yuxin Wu, Laurens van der Maaten, Alan L Yuille, and Kaiming He. Feature denoising for improving adversarial robustness. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 501–509, 2019.

Dong Yin, Raphael Gontijo Lopes, Jonathon Shlens, Ekin D. Cubuk, and Justin Gilmer. A fourier perspective on model robustness in computer vision, 2019.

S. Zagoruyko and N. Komodakis. Wide Residual Networks. *arXiv: 1605.07146*, 2016. URL https://arxiv.org/abs/1605.07146.

A Appendix

Here we present additional technical details for the CIFAR10 and ImageNet experiments.

CIFAR10. We trained all of the models using Adam (Kingma & Ba, 2015) at a base learning rate of 10^-3. We lowered the learning rate three times by a factor of 0.3 each time. The only additional trick to train the CIFAR10 models was to start with $\rho = 100$, anneal down to $\rho = 10$ over 2 epochs, and then anneal to the target ρ over one epoch once training exceeded a threshold of 20%. This jump-start method is inspired by experiments on VIB in Wu et al. (2019). It makes it much easier to train models at low ρ, and appears to not negatively impact final performance.

For the 62×7 models, we used the data augmentation policies for CIFAR-10 found by AutoAug and trained the models for 800 epochs, lowering the learning rate by a factor of 10 at 400 and 600 epochs.

ImageNet. We follow the learning rate schedule for the ResNet 50 from Cubuk et al. (2018), which has a top learning rate of 1.6, trains for 270 epochs, and drops the learning rate by a factor of 10 at 90, 180, and 240 epochs. The only difference for all of our models is that we train at a batch size of 8192 rather than 4096. Similar to the CIFAR10 models, in order to ensure that the ImageNet models train at low ρ, we employ a simple jump-start. We start at $\rho = 100$ and anneal down to the target ρ over 10,000 steps. The first learning rate drop occurs a bit after 14,000 steps. Also similar to the CIFAR 28×10 WRN experiments, none of the models we trained at $\rho = 0$ succeeded, indicating that ResNet 50 and WRN 50×2 both have insufficient capacity to fully learn ImageNet.