Adversarially Trained Models with Test-Time Covariate Shift Adaptation

Jay Nandy\textsuperscript{1}  Sudipan Saha\textsuperscript{2}  Wynne Hsu\textsuperscript{1}  Mong Li Lee\textsuperscript{1}  Xiao Xiang Zhu\textsuperscript{2}

Abstract

We empirically demonstrate that test-time adaptive batch normalization, which re-estimates the batch-normalization statistics during inference, can provide $\ell_2$-certification as well as improve the commonly occurring corruption robustness of adversarially trained models while maintaining their state-of-the-art empirical robustness against adversarial attacks. Furthermore, we obtain similar $\ell_2$-certification as the current state-of-the-art certification models for CIFAR-10 by learning our adversarially trained model using larger $\ell_2$-bounded adversaries. Therefore our work is a step towards bridging the gap between the state-of-the-art certification and empirical robustness. Our results also indicate that improving the empirical adversarial robustness may be sufficient as we achieve certification and corruption robustness as a by-product using test-time adaptive batch normalization.

1. Introduction

Deep neural network (DNN) based models perform poorly in adverse real-world conditions when we receive the test examples from a different distribution than the training examples (Cariucci et al., 2017; Madry et al., 2018; Hendrycks et al. 2019; Nandy et al., 2020). Recent studies found that a DNN classifier that correctly classifies an image $x$, can be easily fooled by an adversarial attack to misclassify $x + \delta$ (Szegedy et al., 2014; Goodfellow et al., 2015; Madry et al., 2018). Here, $\delta$ is a minor adversarial perturbation such that the change between $x$ and $x + \delta$ remains imperceptible to the human eye. Further, DNN classifiers are also sensitive to naturally occurred random corruptions (Hendrycks and Dietterich, 2019; Geirhos et al., 2019). Hence, developing a robust classification framework tackling both adversarial and corruption robustness has emerged as a significant research problem for real-world applications (Gilmer et al., 2019).

Among the successful defense frameworks against adversarial attacks, adversarial training provides the best empirical robustness for a specific perturbation type (such as a small $\ell_p$-norm) by training them using adversarial examples from the same perturbation type (Madry et al., 2018; Tramèr and Boneh, 2019; Zhang et al., 2019; Rice et al., 2020; Gowal et al., 2020). Several certification techniques have been proposed for adversarially trained models (AT models) that attempt to verify if the prediction of a test example, $x$ remains
constant within its neighborhood (Wong and Kolter, 2018; Wang et al., 2018; Salman et al., 2019b; Dvijotham et al., 2018; Gehr et al. 2018; Sheikholeslami et al., 2021). However, these certification techniques typically do not scale for larger networks (e.g., ResNet50) and datasets (e.g., ImageNet). In contrast to adversarial training, randomized smoothing technique provides a scalable $\ell_2$-certification for any classification model that is robust against standard isotropic Gaussian noise (Cohen et al., 2019; Salman et al., 2019a). However, the existing randomized smoothing-based certification models produce poor empirical robustness against adversarial attacks compared to the AT models. Consequently, this certification techniques cannot be applied for AT models as they are not robust against such Gaussian augmented noise. Further, while Gilmer et al. (2019) recently suggest a close relation between adversarial and corruption robustness, AT models often performance even poorly compared to the standard (non-robust) DNN models when exposed to common corruptions (e.g., ImageNet-C) (Hendrycks and Dietterich, 2019), limiting their practical purview. Hence, it raises the question: How can we make the AT models useful for real-world applications where the nature of corruption is unknown?

In standard inference setup, existing AT models independently process each test example. However, this standard inference setup often underestimates their robustness in non-IID environments. For example, the distribution of test examples may change in the medical imaging applications as we use a different data acquisition system (Nandy et al., 2016) or in autonomous cars, satellite image analysis as the weather condition changes (Saha et al., 2019b,a). However, these external conditions change progressively for most (although not for all) of the real-world applications. Hence, we can expect to receive potentially a large number of test examples from the same shifted distribution during inference. We can use these unlabeled test examples in an unsupervised manner to adapt our classifier into such covariate shift.

Unsupervised adaptation mechanisms are well studied in the field of domain adaptation (DA), where the aim is to modulate the model parameters for a different target domain (Li et al. 2016; Huang and Belongie, 2017; Tobin et al., 2017; Zhang et al., 2018). Inspired by these DA techniques, recent works also explored test-time self-supervision (Sun et al., 2020) and adaptive batch-normalization (BN) (Schneider et al., 2020; Nado et al., 2020) using test batches to improve the robustness against random corruptions for standard DNN classifiers. Unlike (Schneider et al., 2020, Nado et al., 2020), we apply the adaptive BN technique for AT models to achieve certified robustness, along with improving their corruption robustness while maintaining their empirical performance against adversarial attacks.

We use CIFAR-10 and ImageNet datasets to present the following contributions: First, we show that test-time adaptive BN significantly improves the robustness of an AT model for any large random perturbations, including Gaussian noise. This simple realization enables us to transform an AT model into a randomized smoothing classifier to provide certification for $\ell_2$ norm, even if the model is learned using $\ell_\infty$-bounded adversaries. Further, we achieve state-of-the-art certification for $\ell_2$ norm, (which is comparable (although not outperforming) to the existing state-of-the-art certification by Salman et al. (2019a)) for the CIFAR-10 by learning the AT model with larger $\ell_2$-bounded adversaries. We also maintain the state-of-the-art empirical robustness of AT model against adversarial attacks. Here, we note that existing certification models typically degrade their empirical robustness. Our experimental results also suggest that empirical robustness and certified robustness are more
closely related than previously believed (Cohen et al., 2019; Salman et al., 2019a. Tramèr and Boneh, 2019). Furthermore, we show that AT models with adaptive BN significantly improves the robustness against common corruptions, mitigating the performance gap from clean images. This improves their overall generalization for real-world applications.

In summary, we bridge the gap between state-of-the-art certification (Cohen et al., 2019; Salman et al., 2019a) and empirically robust models (Madry et al., 2018; Rice et al., 2020). We provide the test-time flexibility to obtain empirically robust predictions along with certifying those predictions for sensitive real-world applications. To the best of our knowledge, existing specialized frameworks typically provide either empirical adversarial robustness (Madry et al., 2018; Rice et al., 2020) or certification (Cohen et al., 2019; Salman et al., 2019a) or corruption robustness (Hendrycks and Dietterich, 2019; Sun et al., 2020). Further, it is challenging to combine these models to obtain appropriate predictions without knowing the nature of the inputs (i.e., adversarial/ random corruption) in real-world applications. In contrast, we achieve all of the above from a single AT model by utilizing simple adaptive BN. The simplicity of our proposed mechanism is a major advantage: it can be applied to any network without additional training or architectural overhead.

2. Background and Related Work

Existing defense models against adversarial attacks can be broadly classified into empirical and certified defenses. Empirical defenses provide empirical robustness against adversarial attacks. Adversarial training achieves the state-of-the-art empirical defense (Madry et al., 2018). It optimizes the following loss function for a DNN classifier, \( f \), to provide robustness within an \( \epsilon \)-bounded threat model for an \( \ell_p \) norm, where the perturbations \( \delta \in \Delta \) are constrained as \( \Delta = \{ \delta : ||\delta||_p \leq \epsilon \} \):

\[
\min_{\theta} \mathbb{E}_{(x,y)} \left[ \max_{\delta \in \Delta} \mathcal{L}(f_\theta(x + \delta), y) \right]
\]

where \( \theta \) denotes the model parameters and \( \mathcal{L} \) is the classification loss.

The inner maximization in Eq. 1 can be solved by producing adversarial examples using strong adversaries, e.g., projected gradient descent (PGD) attack (Kurakin et al., 2016; Madry et al., 2018). However, Wong et al. (2020) found that even a single-step fast gradient sign method (FGSM) attack-based AT models (Goodfellow et al., 2015) achieves high empirical robustness. Recently Trades (Zhang et al., 2019), Adv-LLR (Qin et al., 2019) introduced additional regularizers to achieve higher empirical robustness. However, Rice et al. (2020) showed that the standard PGD based AT model with early-stopping criteria provides one of the strongest empirical defenses for a given perturbation type. Recent works also explored the importance of different hyper-parameters of adversarial training (Gowal et al., 2020; Pang et al., 2021) as well as incorporating additional data in a semi-supervised fashion (Carmon et al., 2019; Uesato et al., 2019) to further improve their empirical robustness.

Empirically robust models are shown to be effective against the known adversaries without providing any robustness guarantees. Several empirical defense models are later "broken" by stronger adversaries (Athalye et al., 2018; Uesato et al., 2018; Jalal et al., 2019). Several certified defenses have been proposed to achieve provable robustness guarantees for AT models to a specific perturbation type. However, these techniques typically do not scale
for large networks (e.g., ResNet50) or datasets (e.g., ImageNet) (Tjeng et al., 2017; Wong and Kolter, 2018; Wang et al., 2018; Singh et al., 2018; Mirman et al., 2018; Raghunathan et al., 2018; Salman et al., 2019b; Croce et al., 2019; Weng et al., 2018; Sheikholeslami et al., 2021). While Mueller et al. (2021) recently proposed to combine the robust AT models with a smaller certified network, their success relies on external adaptive selection criteria to choose the appropriate network for each test image. Towards this, randomized smoothing (Cohen et al., 2019) was proposed as a probabilistically certified defense for $\ell_2$ norm to scale up for practical networks.

Randomized smoothing was initially proposed as a heuristic defense (Cao and Gong, 2017; Liu et al., 2018) and later shown to be certifiable (Lecuyer et al., 2019; Li et al., 2019). Recently, Cohen et al. (2019) and Salman et al. (2019a) provided a tight robustness guarantee for randomized smoothing that achieves the state-of-the-art certification $\ell_2$-norm. It transforms an original ‘base’ classifier $f$ into a smoothed classifier $g$. For an input $x$, $g(x)$ labels $x$ as class $y$ that the base classifier is most likely to return under noisy corruption $x + \delta$, that is,

$$g(x) = \arg \max_{y \in \mathcal{Y}} P(f(x + \delta) == y) \tag{2}$$

where, $\mathcal{Y}$ is the set of class labels and $\delta \sim \mathcal{N}(0, \sigma^2 I)$ is sampled from an isotropic Gaussian distribution with standard deviation, $\sigma$.

Suppose a base classifier $f$ classifies $\mathcal{N}(x, \sigma^2 I)$ to return the “most probable” class, $c_A$ with probability $p_A = P(f(x + \delta) == c_A)$ and the “runner-up” class $c_B$ with probability $p_B = \max_{y \neq c_A} P(f(x + \delta) == y)$. Then, the smooth classifier, $g$ is certifiably robust around $x$ within an $\ell_2$ radius of $R$:

$$R = \frac{\sigma}{2} \left( \Phi^{-1}(p_A) - \Phi^{-1}(p_B) \right) \tag{3}$$

where $\Phi$ is the inverse of the standard Gaussian CDF.

However, computing the exact values of $p_A$ and $p_B$ is not possible in practice. Hence, Monte Carlo sampling is used to approximately estimate $p_A$ and $p_B$ such that $p_A \leq p_B$ with arbitrarily high probability. The certified radius for input $x$ is then computed by replacing $p_A$ and $p_B$ with $\overline{p_A}$ and $\overline{p_B}$ respectively. Recently Salman et al. (2019a) proposed adaptive adversarial training to improve the randomized smoothing framework. Recently, Yang et al. (2020) analyzed this framework for other $\ell_p$ norms using different noise distributions. However, AT-models are not robust against Gaussian noise in standard inference setup. Hence, we cannot transform them into a smoothed classifier to certify for $\ell_2$ norm using this technique (Equation 2). Notably, a classifier can be converted to a smoothed classifier using a separate denoising module as a pre-processor (Salman et al., 2020). However, it still requires retraining their denoiser to generalize their model for different perturbation types.

Furthermore, understanding the interplay between adversarial robustness and corruption robustness is a longstanding problem (Gilmer et al., 2019; Tramèr and Boneh, 2019). However, the robustness of AT models does not generalize well to naturally occurring common corruptions (e.g., ImageNet-C) (Gilmer et al., 2019; Yin et al., 2019; Hendrycks et al., 2020a). Several recent works only focused on improving corruption robustness.
that typically involves special training with additional time and resources. These include data augmentation using Gaussian noise (Gilmer et al., 2019), optimized mixtures of data augmentations in conjunction with a consistency loss (Hendrycks et al., 2020b), training on stylized images (Geirhos et al., 2019; Michaelis et al., 2019), or using adversarially generated noises (Rusak et al., 2020). Other approaches tweak the architecture, e.g., by adding shift-equivariance with an anti-aliasing module (Zhang, 2019) or assembling different training techniques (Lee et al., 2020).

Recently, test-time adaptation techniques using self-supervision (Sun et al., 2020) or adaptive BN (Schneider et al., 2020; Nado et al., 2020) were explored to improve corruption robustness for standard (non-robust) DNN classifiers. However, these methods cannot provide both adversarial and corruption robustness for the same classifier. Notably, recent studies also investigated reducing/removing the BN layers (Galloway et al., 2019) and introducing auxiliary BN layers (Xie et al., 2020) to improve the empirical adversarial robustness. In this paper, we propose test-time BN adaptation for AT models (instead of applying for standard DNNs as (Schneider et al., 2020; Nado et al., 2020)) to improve both adversarial and corruption robustness.

3. Adaptive Batch-Normalization

Batch-Normalization is a powerful tool to improve the training stability and attain faster convergence (Ioffe and Szegedy, 2015). A batch-normalization (BN) layer estimates the mean and variance of the hidden activation maps across the channels. The feature activations are then normalized to $\mathcal{N}(0, 1)$ before feeding into the next hidden layer. During training, we compute the means and the variances directly from the training mini-batches. However, in the standard inference setup, test examples are assumed to be sampled from an identical distribution as the training examples. Further, test examples are processed independently from each other. Hence, we also estimate a running mean and variance during training by applying exponential averaging over the training batches to normalize the hidden feature activations of the test examples.

BN statistics estimated from the training batches generalizes well in the standard IID settings. However, in non-IID settings, the shift in the test distribution lead to different activation statistics for the test examples. Hence, impacted by the covariate shift, the statistics estimated using the training batches fail to normalize the activation tensors to $\mathcal{N}(0, 1)$. It breaks the crucial assumption for the subsequent hidden layers to work.

More formally, let $x \in \mathcal{X}$ are inputs and $y \in \mathcal{Y}$ are the class labels. We denote the training distribution as $P_T : \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{R}^+$ and test distribution as $P_t : \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{R}^+$. There exists covariate shift between training and test distribution iff: $P_T(y|x) = P_t(y|x)$ and $P_T(x) \neq P_t(x)$ (Sugiyama and Kawanabe, 2012; Schölkopf et al., 2012). If the covariate shift only leads to change in the first and second-order moments of the feature activations $f_h(x)$, we can remove it by re-estimating these statistics using the test batches, followed by normalization (Schneider et al., 2020):

$$P_T\left(\frac{f_h(x) - \mathbb{E}_T[f_h(x)]}{\sqrt{\mathbb{V}_T[f_h(x)]}}\right)P_T(x) \approx P_t\left(\frac{f_h(x) - \mathbb{E}_t[f_h(x)]}{\sqrt{\mathbb{V}_t[f_h(x)]}}\right)P_t(x).$$  (4)
In other words, we can remove the covariate shift by correcting first and second-order moments as long as the change in the test distribution only leads the feature activations to scale and translate. Modulating the BN statistics adaptive BN is effective for domain adaptation (Li et al., 2016; Saha et al., 2018) or corruption robustness (Nado et al., 2020; Schneider et al., 2020) as long as the semantics in the test images do not change (Li et al., 2016; Nado et al., 2020; Schneider et al., 2020).

Adaptive BN modulates the BN statistics using unlabelled test examples during inference. We compute the BN statistics from the feature activations of the test batch, denoted as $\mu_t$, $s_t^2$. We adapt them with the existing training statistics, $\mu_T$, $s_T^2$, obtained using the training batches as:

$$\bar{\mu} = \rho \cdot \mu_t + (1 - \rho) \cdot \mu_T \quad \bar{s} = \rho \cdot s_t + (1 - \rho) \cdot s_T$$ (5)

where, $\rho \in [0, 1]$ is the momentum.

The choice of $\rho = 0$ rejects the statistics of the test examples. It is equivalent to the standard inference setup with a deterministic DNN classifier in the IID settings. In contrast, $\rho = 1$ completely ignores the pre-calculated statistics from the training batches. As we receive a larger test batch, we can get a better estimation for the test distribution. Hence, we should choose a higher value for $\rho$. However, the choice of $\rho$ often depends on practical constraints, e.g., available hardware resources. Consequently, if the test-batch size is too small, the estimations become unreliable. The appropriate value for $\rho$ can be chosen empirically for different applications, depending on the available test batch size.

4. Experiments

We present four sets of experiments to demonstrate the effectiveness of the test-time adaptive BN technique for AT models.

4.1 Experimental Setup

We use CIFAR-10 (Krizhevsky et al., 2009) and ImageNet (Deng et al., 2009) for our experiments. We use pre-activation ResNet18 for CIFAR-10, and ResNet50 for ImageNet (He et al., 2016a,b). Unless otherwise specified, our AT models are trained using early stopping criteria (Rice et al., 2020) as follows: For ImageNet, the AT models Adv$_\infty$ and Adv$_2$ are learned for $\ell_\infty$ and $\ell_2$ threat models with threat boundaries of 4/255 and 3 respectively. For CIFAR-10, we mainly experiment with two AT models, Adv$_\infty$ and Adv$_2$, for $\ell_\infty$ and $\ell_2$ threat models with threat boundaries of 8/255 and 1, respectively. We compare with Baseline and Rand$_{\sigma=0.5}$ models that are Our baseline model is trained using clean images. Rand$_{\sigma=0.5}$ is trained by augmenting random noise, sampled from isotropic Gaussian distribution $\mathcal{N}(0, \sigma^2I)$ with $\sigma = 0.5$ for both datasets. We also compare with the state-of-the-art certification models by Salman et al. (2019a) for our $\ell_2$ certification experiments with CIFAR-10. Please refer to Appendix A for more details.

1. For ImageNet, we obtain Adv$_\infty$ and Adv$_2$ from https://github.com/locuslab/robust_overfitting and Baseline and Rand$_{\sigma=0.5}$ models from https://github.com/locuslab/smoothing.
Table 1: Top-1 accuracy of different classifiers under different levels of Gaussian noises augmented to the test images. We randomly shuffle test images and sample the noises and report (mean ± 2 × sd) of five runs. Results for our proposed method are shown in gray.

| Model          | σ = 0 | σ = 0.25 | σ = 0.5 | σ = 0.75 |
|----------------|-------|----------|---------|----------|
| Baseline       | 73.2±0.6 | 11.8±0.25 | 0.4±0.03 | 0.1±0.04 |
| + adaptive BN  | 74.4±0.6 | 31.0±0.25 | 7.7±0.04 | 2.4±0.04 |
| Adv∞           | 62.8±0.6 | 3.9±0.03  | 0.4±0.03 | 0.2±0.01 |
| + adaptive BN  | 60.8±0.6 | 53.4±0.25 | 44.9±0.03| 33.7±1.25|
| Adv2           | 59.8±0.6 | 9.8±0.25  | 0.9±0.03 | 0.3±0.06 |
| + adaptive BN  | 58.3±0.6 | 53.7±0.25 | 47.3±0.03| 39.8±1.25|
| Randσ=0.5     | 22.0±0.6 | 32.8±0.25 | 60.9±0.03 | 9.3±0.06 |
| + adaptive BN  | 62.7±0.6 | 62.3±0.25 | 59.5±0.03| 51.4±0.27 |

Figure 1: Visualizing loss-gradients produced by AT models as we apply different levels of Gaussian noises.

4.2 Robustness against Gaussian Noise

In Table 1, we present the performance of different models under different levels of Gaussian noises with and without applying test-time adaptive BN. We observe that when the test examples are sampled from IID settings (i.e., σ = 0 for Baseline, Adv∞, and Adv2 and σ = 0.5 for Randσ=0.5), the performance of the models remain similar regardless of whether BN adaptation is applied. However, as we move away from the IID settings by increasing (or decreasing) σ, the performance of these models significantly degrades in the standard inference setup. In contrast, test-time adaptive BN improves the performance for all classification models (Schneider et al., 2020; Nado et al., 2020).

However, AT models achieve significantly higher performance gain using adaptive BN compared to the standard non-robust models. For example, at σ = 0.5, the baseline, Adv2 and Adv∞ respectively achieve top-1 accuracy of 0.3%, 0.4%, and 0.9% for ImageNet without using BN adaptation (Table 1 (a)). However, by applying test-time adaptive BN, Adv2 and
Adv$_\infty$ significantly improves the top-1 accuracy to 44.9%, 47.3% respectively. In contrast, after correcting the BN statistics, the Baseline achieves significantly lower top-1 accuracy of only 7.7%. We also observe similar results for CIFAR-10 in Table 1 (b).

In Figure 1, we visualize the *loss gradients* for individual pixels of an image as we increase the Gaussian noise (i.e., $\sigma$) to understand the effect of test-time adaptive BN for AT models. *Loss-gradients* reflect the most relevant input pixels for classification predictions. Here, we scale, translate and clip the loss-gradient values for better visualization (as in (Tsipras et al., 2019)), without applying complex processing techniques. At $\sigma = 0$ (i.e., for clean images), the loss-gradients from AT models align properly with perceptually relevant features (as observed previously (Tsipras et al. 2019; Etmann et al. 2019)). On the other hand, as we increase the noise level to $\sigma = 0.5$ and $\sigma = 0.75$, the overall loss gradient becomes noisier. However, the AT models without BN adaptation produce sharper loss gradients (i.e., providing higher importance) even for background pixels. In contrast, test-time BN adaptation allows us to obtain sharper loss gradients for the pixels from the object of interest while suppressing the background pixels (see Figure 1(c) and Figure 1(d)). Hence, the AT models extracts the required semantic information for correct classifications. It is interesting to note that Adv$_2$ produces significantly more human-aligned loss gradients compared to Adv$_\infty$. This behavior is also reflected in their classification as we can see that Adv$_2$ achieves higher top-1 accuracy compared to Adv$_\infty$ (Table 1). Further in Appendix D.2, we also observe similar results for other random corruptions from ImageNet-C. Hence, we need a minor test-time adjustment to the learned weights to make the AT models robust against random covariate shifts.

### 4.3 Certification using Randomized Smoothing

A classifier, $f$ that achieves robustness against standard Gaussian noise, can be transformed into a randomized smoothed classifier, $g$ (Equation 2) to provide certified robustness against $\ell_2$-norm (Equation 3). Since modulating BN statistics using test-time adaptive BN technique significantly improves robustness against Gaussian noise for both Adv$_2$ and Adv$_\infty$, we
can transform both of these AT models into a smoothed classifier to provide robustness certification for $\ell_2$ norm.  

For a given test batch, we provide the certification as follows: We first apply adaptive BN to modulate the BN statistics of our classifier using the same test batch, augmented with Gaussian noises. The noises should be sampled from the same isotropic Gaussian distribution $\mathcal{N}(0, \sigma^2 I)$ as used for certification. Then, we freeze the model parameters to certify that test batch and use the adapted model as our base classifier $f$. Hence, the base classifier $f$ remains the same to calculate the certification radius $R$ (Equation 3).

Certified accuracy at radius $R$ is defined as the fraction of test examples that the smoothed classifier, $g$ classifies correctly without abstaining and certifies as robust for at least an $\ell_2$ radius of $R$. We estimate the class label probabilities of $g$ using Monte-Carlo sampling with 100,000 noisy samples for each test image, as in (Cohen et al., 2019; Salman et al., 2019a). We certify the test images with 99.9% probability. That is, there is at most a 0.1% chance that the certification technique falsely certifies a non-robust test example.

In Figure 2, we present the results of certified accuracy. We use the full test-set for CIFAR-10 and a subsampled of 500 test images for ImageNet (as in (Cohen et al., 2019)). We use $\sigma = 0.5$ for Adv$_{\infty}$ and Adv$_2$ with test-time BN adaptation for both datasets. For the Baseline, Adv$_{\infty}$ and Adv$_2$ models without BN adaption, we use $\sigma = 0.25$ that are not robust against Gaussian noises (see Table 1). We can see that both Adv$_{\infty}$ and Adv$_2$ models with adaptive BN provide certified robustness for $\ell_2$ norm. We note that Adv$_2$ models consistently achieve better performance compared to Adv$_{\infty}$ in terms of certified accuracy. For CIFAR-10, both Adv$_{\infty}$ and Adv$_2$ models with test-time adaptive BN outperform the standard randomized smoothing framework i.e., Rand$_{\sigma=0.5}$ at $\sigma = 0.5$ (Cohen et al., 2019). For ImageNet, Adv$_2$ achieves higher certified accuracy compared to Rand$_{\sigma=0.5}$ beyond $\ell_2$-radii of 1.5. Note that, adaptive BN allows the flexibility of choosing $\sigma$ to set the appropriate noise level for the certification process. Hence, we may use different $\sigma$ values to certify different test examples using the same robust network, unlike (Cohen et al., 2019; Salman et al., 2019a) (Please refer to Appendix B.1 for additional experiments).

An $\ell_2$ ball of radius $\sqrt{d}$ contains an $\ell_\infty$ unit ball in $\mathbb{R}^d$. Hence, we can use this naive conversion to certify our Adv$_{\infty}$ and Adv$_2$ models with BN adaptation for $\ell_\infty$ norm (as in (Salman et al., 2019a)). For CIFAR-10 at $\ell_\infty$ radius of $2/255$ (i.e., $\ell_2$ certified accuracy at $2\sqrt{2}^2 \times 3/255 \approx 0.435$), our Adv$_{\infty}$ and Adv$_2$ provide certified accuracy of 45.8% and 49.0, compared to 43.0% for Rand$_{\sigma=0.5}$. For ImageNet at $\ell_\infty$ radius of $1/255$ (i.e $\ell_2$ certified accuracy at $2\sqrt{224^2 \times 3}/255 \approx 1.5$), our Adv$_{\infty}$ and Adv$_2$ provide certified accuracy of 23.4% and 30.8%, compared to 29.0% for Rand$_{\sigma=0.5}$.

We note that Salman et al. (2019a) improved randomized smoothing models to achieve the current state-of-the-art certified accuracy for $\ell_2$-norm. However, these randomized-smoothing models (Cohen et al., 2019; Salman et al., 2019a) perform poorly against adversarial attacks compared to the AT models (Rice et al., 2020; Gowal et al., 2020). Further, all of these existing adversarially robust models typically perform poorly against most common corruptions (Gilmer et al., 2019). In contrast, by applying the adaptive BN for AT models, we are providing $\ell_2$-certification as well as corruption robustness (section 4.5) without degrading their empirical robustness (section 4.4). Moreover in Figure 3(a) and Figure 3(b), we

---

2. Please refer to Table 6 and Table 7 in Appendix B for certified accuracy values corresponding to the plots and results for additional models.
Figure 3: CIFAR-10: Certified top-1 accuracy achieved by (a) Adv$_\infty$ and (b) Adv$_2$ models (with test-time adaptive BN at $\sigma = 0.5$), learned at different threat boundaries. (c) Comparison with the state-of-the-art SmoothAdv models (Salman et al. 2019a), trained at $\sigma = 0.5$ using preactivation ResNet-18 architecture.

Figure 4: CIFAR-10: Comparing the certified top-1 accuracy of Adv$_\infty$ (Left) and Adv$_2$ (Right) models learned using different threat boundaries with and without applying early-stopping criteria (denoted as Adv$_\text{overfit}$).

demonstrate that by learning the Adv$_\infty$ and Adv$_2$ models using larger threat boundaries, we can achieve higher certified accuracy for CIFAR-10.

Figure 3(c) also compare the certified accuracy of Adv$_2$ models with the existing state-of-the-art ‘SmoothAdv’ models by Salman et al. (2019a). SmoothAdv utilizes adversarial training using an adaptive attack with $\ell_2$ threat boundary of $\epsilon$ and Gaussian noises, $\mathcal{N}(0, \sigma^2 I)$ (See details in Appendix A). We set the noise to $\sigma = 0.5$ and vary $\epsilon$ to compare with different SmoothAdv models in Figure 3(c). We observe that both SmoothAdv and Adv$_2$ with BN adaptation achieve similar certified accuracy as we carefully adjust the threat boundaries. See Appendix B.2 for additional results. 3

Rice et al. (2020) recently demonstrate that AT models overfit as we train without early stopping criteria. It degrades their empirical robustness against adversarial attacks. Here, we compare with the certification accuracies of such overfitted AT models, denoted as Adv$_\text{overfit}$. In Figure 4, we further note that they also lead to poor certified robustness, 3

3. We do not compare with Salman et al. (2019a) as adversarial training for ImageNet is still at a nascent stage.
Adversarially Trained Models with Test-Time Covariate Shift Adaptation

### Table 2: Empirical performance of AT models under adversarial attacks.

For models with BN adaptation, we randomly shuffle the test-batches to report (mean + 2 × sd) of 3 different runs. Models are as described in section 4.1. Models without BN adaptation are corresponding to Rice et al. (2020).

![Table 2](image)

in particular, for higher \(\ell_2\) radii, compared to their corresponding AT models with early stopping criteria. This experimental result demonstrates that empirical and certifiable robustness are more closely related than previously believed: improving empirical robustness also leads to improve the certification performance for AT models.

So far, we only focus on \(\ell_2\) certification using Gaussian augmented noise. However, we can also adapt these models to any random noise distribution to provide non-trivial certifications for other \(\ell_p\) norms (e.g., uniform noise for \(\ell_1\) threat models (Yang et al., 2020)), without any additional training (unlike (Salman et al., 2020)). For example, in Section 4.5, we demonstrate that AT models with test-time adaptive BN significantly improves the top-1 accuracy against random corruption images.

### 4.4 Performance under adversarial attacks

AT models provide the best empirical defense against adversarial attacks (Madry et al., 2018; Rice et al., 2020; Gowal et al., 2020; Pang et al., 2021). This section demonstrates that test-time adaptive BN for AT models maintains their empirical robustness against adversarial attacks while providing certification and improve corruption robustness.

For a test image, \(x\) with label \(y\), the goal of an adversarial attack is to find perturbation \(\delta\) within a chosen perturbation \(\epsilon\) for an \(\ell_p\) norm such that:

\[
\max_{\delta \in \Delta} \mathcal{L}(f_{\theta}(x + \delta), y) \quad \text{where} \quad \Delta = \{\delta : ||\delta||_p \leq \epsilon\}
\]  

We use projected gradient descent (PGD) attack (Kurakin et al., 2017; Madry et al., 2018) to evaluate Adv\(_\infty\) and Adv\(_2\) models in the standard inference setup (i.e., without using adaptive BN). It starts with a random initial perturbation, \(\delta^{(0)}\). The perturbation is iteratively adjusted by following the gradient steps with step-size \(\eta\) and projecting it back onto the \(\ell_p\) threat model:

\[
\tilde{\delta} = \delta^{(t)} + \eta \cdot Q_p(\nabla_\delta \mathcal{L}(f_{\theta}(x + \delta^{(t)}), y)), \quad \delta^{(t+1)} = \max(\min(\tilde{\delta}, \epsilon), -\epsilon)
\]

For \(\ell_\infty\), we use the sign of the gradient i.e., \(Q_p(z) = \text{sign}(z)\). For other (finite) norms, such as, \(\ell_2\), we normalize the gradient as: \(Q_p(z) = z/||z||_2\).

In contrast to the standard inference setup, test-time adaptive BN consistently modulates the BN parameters to accommodate the changes in the input examples at each forward pass. Hence, we cannot effectively compute the gradients, \(\nabla_\delta \mathcal{L}(f_{\theta}(x + \delta), y)\) using Equation 7 to
Table 3: Effect of adaptive BN on AT models against corruption images. For models with BN adaptation, we randomly shuffle the test-batches to report (mean + 2 × sd) of 3 different runs. For each column, down-arrow (↓) denotes the lower values are better and up-arrow (↑) denotes the higher values are better.

production the most effective adversarial examples. Instead, we apply the expectation over transformation (EoT) technique to approximate the gradient (Athalye et al., 2017). For cross-entropy loss function, we can write the loss as: ∇δL(fθ(x + δ), y) = ∇δ(−log(fθ(x + δ)y)). To apply EoT, we select a set of m models with varying BN statistics and approximate the gradient as:

\[ \nabla_\delta[−\log E_{\theta\in\Theta}f_\theta(x + \delta)y] \approx \nabla_\delta[−\log \frac{1}{m} \sum_{i=1}^{m} f_{\theta(i)}(x + \delta)y] \quad (8) \]

where, \( \{f_{\theta(i)}\}_{i=1}^{m} \) are m models with different BN parameters.

Each \( f_{\theta(i)} \) is obtained by applying adaptive BN to the original model, \( f_\theta \) using test batch perturbed with randomly chosen \( \delta \in \Delta \). Notably, we place the ‘log’ outside of the ‘expectation’ as it was found to be more effective in (Salman et al., 2019a).

Table 2 compares the performance of AT models as we apply test-time adaptive BN. For our evaluations, we use the same threat boundaries for attacks as used during the adversarial training. We generate the adversarial examples using 100 iterations. We set \( m = 10 \) for AT models using test-time adaptive BN. We do not observe a significant difference by choosing larger \( m \). We use a subsampled test examples of 2000 images for ImageNet while use the entire testset for CIFAR-10.

In Table 2, we see that adaptive BN produces a consistent performance improvement for the ImageNet models. In contrast, it does not impact the performance significantly for CIFAR-10. Overall, test-time adaptive BN for AT models produces no detrimental effect. In Appendix C.2, we further investigate the performance gains as we learn the AT models without early stopping criteria.

4.5 Performance against Common Corruptions

Hendrycks and Dietterich (2019) recently studied robustness of classifiers against different common (non-adversarial) visual corruptions. The authors introduced ImageNet-C and
CIFAR10-C datasets by algorithmically generated random corruptions from noise, blur, weather, and digital categories with 5 different severity levels for each corruption. While AT models are effective against adversarial attacks, they often perform poorly against such corruptions, even compared to the standard neural networks in the standard inference setup (Hendrycks et al., 2020a; Yin et al., 2019; Gilmer et al., 2019). Here, we demonstrate that by applying test-time adaptive BN, we can significantly improve their performance.

The corruption robustness of a classification model, $f$, is typically measured using the mean corruption error ($mCE$) (Hendrycks and Dietterich, 2019). It is computed by normalizing the top-1 errors of $f$ with the top-1 errors for AlexNet model (Krizhevsky et al., 2012) across the set of corruptions, $K$ and corruption severities, $I$, as:

$$mCE(f) = \frac{1}{K} \sum_{k=1}^{K} \sum_{i=1}^{I} \frac{err_{k,i}^f}{err_{k,i}^{\text{Alex}}},$$

where, $err_{k,i}^f$ and $err_{k,i}^{\text{Alex}}$ denotes the top-1 error against corruption type $k$ at severity $i$ for $f$ and the AlexNet model respectively.

However, $mCE$ does not reflect the generalization gap between clean and corruption images. For example, a classifier may withstand the corruption images, minimizing the generalization gaps from the clean test set. However, it may still obtain a higher $mCE$ compared to a classifier with a low clean error rate while the error rate spike in the presence of corruption. Towards this, Hendrycks and Dietterich (2019) proposed another metric, called relative mean corruption error ($rmCE$). It computes a normalized relative change of top-1 error rate of the classifier, $f$ from the clean error rate across the corruptions, $K$ and the corruption severity levels, $I$, as follows:

$$rmCE(f) = \frac{1}{K} \sum_{k=1}^{K} \sum_{i=1}^{I} \frac{err_{k,i}^f - err_{k,i}^{\text{Clean}}}{err_{k,i}^{\text{Alex}} - err_{k,i}^{\text{Clean}}},$$

where, $err_{\text{Clean}}^f$ and $err_{\text{Clean}}^{\text{Alex}}$ are the clean error rate of $f$ and the AlexNet model respectively.

Since AT models produce a high error rate for the clean test images to achieve the adversarial robustness, we argue that $rmCE$ is a more appropriate metric.

Table 3 compares the performance of different classification models against the corruption images. We also report the mean top-1 accuracy for clean and corrupted test images and the performance gaps along with $mCE$ and $rmCE$ scores. We note that AT models already produce lower $rmCE$ scores. Further, by applying test-time adaptive BN, they consistently improve the $rmCE$ scores for both datasets, leading to a better classification generalization. We also note that adaptive BN for AT models significantly reduces the top-1 accuracy gap between clean and corrupted test images for both datasets. In Appendix D.1, we present the performance under individual corruptions. In Appendix D.2, we also visualize the effect of the adaptive BN technique for AT models on the loss gradients under various corruptions and analyze them from a Fourier perspective.

5. Conclusion

In this paper, we show that test-time adaptive BN for an AT model produces certified accuracy as well as corruption robustness while maintaining their state-of-the-art empirical
robustness against adversarial attacks. Further, we achieve the state-of-the-art certification for CIFAR-10 by learning the AT models using higher $\ell_2$ threat boundaries. These results indicate a close relation between empirical and certified robustness than previously believed.

References

Anish Athalye, Logan Engstrom, Andrew Ilyas, and Kevin Kwok. Synthesizing robust adversarial examples. arXiv, 2017.

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

Xiaoyu Cao and Neil Zhenqiang Gong. Mitigating evasion attacks to deep neural networks via region-based classification. In ACSAC, 2017.

Fabio Maria Cariucci, Lorenzo Porzi, Barbara Caputo, Elisa Ricci, and Samuel Rota Bulo. Autodial: Automatic domain alignment layers. In ICCV, 2017.

Yair Carmon, Aditi Raghunathan, Ludwig Schmidt, Percy Liang, and John C Duchi. Unlabeled data improves adversarial robustness. arXiv, 2019.

Jeremy M Cohen, Elan Rosenfeld, and J Zico Kolter. Certified adversarial robustness via randomized smoothing. ICML, 2019.

Francesco Croce, Maksym Andriushchenko, and Matthias Hein. Provable robustness of relu networks via maximization of linear regions. In AISTATS, 2019.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In CVPR, 2009.

Krishnamurthy Dvijotham, Robert Stanforth, Sven Gowal, Timothy A Mann, and Pushmeet Kohli. A dual approach to scalable verification of deep networks. In UAI, 2018.

Logan Engstrom, Andrew Ilyas, Hadi Salman, Shibani Santurkar, and Dimitris Tsipras. Robustness (python library), 2019. URL https://github.com/MadryLab/robustness.

Christian Etmann, Sebastian Lunz, Peter Maass, and Carola-Bibiane Schönlieb. On the connection between adversarial robustness and saliency map interpretability. In ICML, 2019.

Angus Galloway, Anna Golubeva, Thomas Tanay, Medhat Moussa, and Graham W Taylor. Batch normalization is a cause of adversarial vulnerability. arXiv, 2019.

Timon Gehr, Matthew Mirman, Dana Drachsler-Cohen, Petar Tsankov, Swarat Chaudhuri, and Martin Vechev. Ai2: Safety and robustness certification of neural networks with abstract interpretation. In IEEE SP, 2018.

Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A Wichmann, and Wieland Brendel. Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. In ICLR, 2019.
Justin Gilmer, Nicolas Ford, Nicholas Carlini, and Ekin Cubuk. Adversarial examples are a natural consequence of test error in noise. In *ICML*, 2019.

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

Sven Gowal, Chongli Qin, Jonathan Uesato, Timothy Mann, and Pushmeet Kohli. Unering the limits of adversarial training against norm-bounded adversarial examples. *arXiv*, 2020.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE CVPR*, 2016a.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *ECCV*, 2016b.

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

Dan Hendrycks, Mantas Mazeika, and Thomas Dietterich. Deep anomaly detection with outlier exposure. In *ICLR*, 2019.

Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, Dawn Song, Jacob Steinhardt, and Justin Gilmer. The many faces of robustness: A critical analysis of out-of-distribution generalization. *arXiv*, 2020a.

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. *ICLR*, 2020b.

Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *CVPR*, 2017.

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

Ajit Jalal, Andrew Ilyas, Constantinos Daskalakis, and Alexandros G Dimakis. The robust manifold defense: Adversarial training using generative models. *arXiv*, 2019.

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

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *NeurIPS*, 2012.

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

Alexey Kurakin, Ian J. Goodfellow, and Samy Bengio. Adversarial examples in the physical world. In *Workshop track, ICLR*, 2017.
Mathias Lecuyer, Vaggelis Atlidakis, Roxana Geambasu, Daniel Hsu, and Suman Jana. Certified robustness to adversarial examples with differential privacy. In IEEE SP, 2019.

Jungkyu Lee, Taeryun Won, Tae Kwan Lee, Hyemin Lee, Geonmo Gu, and Kiho Hong. Compounding the performance improvements of assembled techniques in a convolutional neural network. arXiv, 2020.

Bai Li, Changyou Chen, Wenlin Wang, and Lawrence Carin. Certified adversarial robustness with additive noise. NeurIPS, 2019.

Yanghao Li, Naiyan Wang, Jianping Shi, Jiaying Liu, and Xiaodi Hou. Revisiting batch normalization for practical domain adaptation. arXiv, 2016.

Xuanqing Liu, Minhao Cheng, Huan Zhang, and Cho-Jui Hsieh. Towards robust neural networks via random self-ensemble. In ECCV, 2018.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In ICLR, 2018.

Claudio Michaelis, Benjamin Mitzkus, Robert Geirhos, Evgenia Rusak, Oliver Bringmann, Alexander S Ecker, Matthias Bethge, and Wieland Brendel. Benchmarking robustness in object detection: Autonomous driving when winter is coming. arXiv, 2019.

Matthew Mirman, Timon Gehr, and Martin Vechev. Differentiable abstract interpretation for provably robust neural networks. In ICML, 2018.

Mark Niklas Mueller, Mislav Balunovic, and Martin Vechev. Certify or predict: Boosting certified robustness with compositional architectures. In ICLR, 2021.

Zachary Nado, Shreyas Padhy, D Sculley, Alexander D’Amour, Balaji Lakshminarayanan, and Jasper Snoek. Evaluating prediction-time batch normalization for robustness under covariate shift. arXiv, 2020.

Jay Nandy, Wynne Hsu, and Mong Li Lee. An incremental feature extraction framework for referable diabetic retinopathy detection. In ICTAI. IEEE, 2016.

Jay Nandy, Wynne Hsu, and Mong-Li Lee. Towards maximizing the representation gap between in-domain & out-of-distribution examples. In NeurIPS, 2020.

Tianyu Pang, Xiao Yang, Yinpeng Dong, Hang Su, and Jun Zhu. Bag of tricks for adversarial training. In International Conference on Learning Representations, 2021. URL https://openreview.net/forum?id=Xb8xvrtB8Ce

Chongli Qin et al. Adversarial robustness through local linearization. In NeurIPS, 2019.

Aditi Raghunathan, Jacob Steinhardt, and Percy Liang. Semidefinite relaxations for certifying robustness to adversarial examples. NeurIPS, 2018.

Leslie Rice, Eric Wong, and Zico Kolter. Overfitting in adversarially robust deep learning. In ICML, 2020.
Evgenia Rusak, Lukas Schott, Roland Zimmermann, Julian Bitterwolf, Oliver Bringmann, Matthias Bethge, and Wieland Brendel. Increasing the robustness of dnns against image corruptions by playing the game of noise. *arXiv*, 2020.

Sudipan Saha, Francesca Bovolo, and Lorenzo Bruzzone. Destroyed-buildings detection from vhr sar images using deep features. In *Image and Signal Processing for Remote Sensing*, 2018.

Sudipan Saha, Francesca Bovolo, and Lorenzo Bruzzone. Unsupervised deep change vector analysis for multiple-change detection in vhr images. *IEEE Transactions on Geoscience and Remote Sensing*, 2019a.

Sudipan Saha, Francesca Bovolo, and Lorenzo Bruzzone. Unsupervised multiple-change detection in vhr multisensor images via deep-learning based adaptation. In *IEEE International Geoscience and Remote Sensing Symposium*, 2019b.

Hadi Salman, Jerry Li, Ilya Razenshteyn, Pengchuan Zhang, Huan Zhang, Sebastien Bubeck, and Greg Yang. Provably robust deep learning via adversarially trained smoothed classifiers. In *NeurIPS*, 2019a.

Hadi Salman, Greg Yang, Huan Zhang, Cho-Jui Hsieh, and Pengchuan Zhang. A convex relaxation barrier to tight robustness verification of neural networks. In *NeurIPS*, 2019b.

Hadi Salman, Mingjie Sun, Greg Yang, Ashish Kapoor, and J Zico Kolter. Denoised smoothing: A provable defense for pretrained classifiers. *NeurIPS*, 2020.

Steffen Schneider, Evgenia Rusak, Luisa Eck, Oliver Bringmann, Wieland Brendel, and Matthias Bethge. Improving robustness against common corruptions by covariate shift adaptation. *NeurIPS*, 2020.

Bernhard Schölkopf, Dominik Janzing, Jonas Peters, Eleni Sgouritsa, Kun Zhang, and Joris Mooij. On causal and anticausal learning. *arXiv preprint arXiv:1206.6471*, 2012.

Fatemeh Sheikholeslami, Ali Lotfi, and J Zico Kolter. Provably robust classification of adversarial examples with detection. In *ICLR*, 2021.

Gagandeep Singh, Timon Gehr, Matthew Mirman, Markus Püschel, and Martin Vechev. Fast and effective robustness certification. In *NeurIPS*, 2018.

Masashi Sugiyama and Motoaki Kawanabe. *Machine learning in non-stationary environments: Introduction to covariate shift adaptation*. MIT press, 2012.

Yu Sun, Xiaolong Wang, Zhuang Liu, John Miller, Alexei Efros, and Moritz Hardt. Test-time training with self-supervision for generalization under distribution shifts. In *ICML*, 2020.

Christian Szegedy et al. Intriguing properties of neural networks. In *ICLR*, 2014.

Vincent Tjeng, Kai Xiao, and Russ Tedrake. Evaluating robustness of neural networks with mixed integer programming. *ICLR*, 2017.
Josh Tobin, Rachel Fong, Alex Ray, Jonas Schneider, Wojciech Zaremba, and Pieter Abbeel. Domain randomization for transferring deep neural networks from simulation to the real world. In IROS, 2017.

Florian Tramèr and Dan Boneh. Adversarial training and robustness for multiple perturbations. In NeurIPS, 2019.

Dimitris Tsipras, Shibani Santurkar, Logan Engstrom, Alexander Turner, and Aleksander Madry. Robustness may be at odds with accuracy. In ICLR, 2019.

Jonathan Uesato, Brendan O’Donoghue, Aaron van den Oord, and Pushmeet Kohli. Adversarial risk and the dangers of evaluating against weak attacks. ICML, 2018.

Jonathan Uesato, Jean-Baptiste Alayrac, Po-Sen Huang, Robert Stanforth, Alhussein Fawzi, and Pushmeet Kohli. Are labels required for improving adversarial robustness? arXiv, 2019.

Shiqi Wang, Kexin Pei, Justin Whitehouse, Junfeng Yang, and Suman Jana. Efficient formal safety analysis of neural networks. In NeurIPS, 2018.

Lily Weng, Huan Zhang, Hongge Chen, Zhao Song, Cho-Jui Hsieh, Luca Daniel, Duane Boning, and Inderjit Dhillon. Towards fast computation of certified robustness for relu networks. In ICML, 2018.

Eric Wong and Zico Kolter. Provable defenses against adversarial examples via the convex outer adversarial polytope. In ICML, 2018.

Eric Wong, Leslie Rice, and J Zico Kolter. Fast is better than free: Revisiting adversarial training. ICLR, 2020.

Cihang Xie, Mingxing Tan, Boqing Gong, Jiang Wang, Alan L Yuille, and Quoc V Le. Adversarial examples improve image recognition. In IEEE CVPR, 2020.

Greg Yang, Tony Duan, Edward Hu, Hadi Salman, Ilya Razenshteyn, and Jerry Li. Randomized smoothing of all shapes and sizes. arXiv, 2020.

Dong Yin, Raphael Gontijo Lopes, Jon Shlens, Ekin Dogus Cubuk, and Justin Gilmer. A fourier perspective on model robustness in computer vision. In NeurIPS, 2019.

Hang Zhang, Kristin Dana, Jianping Shi, Zhongyue Zhang, Xiaogang Wang, Ambrish Tyagi, and Amit Agrawal. Context encoding for semantic segmentation. In CVPR, 2018.

Hongyang Zhang et al. Theoretically principled trade-off between robustness and accuracy. In ICML, 2019.

Richard Zhang. Making convolutional networks shift-invariant again. In ICML, 2019.
Appendix Organization

We organize the appendix as follows:

- **Section A: Experimental setup.**
  - In section A.1, we present the implementation details for training the classification models.
  - Section A.2 discusses the choice of test-time hyper-parameter settings for using the test-time adaptive batch-normalization (BN) technique.

- **Section B: Additional Results on Certification.**
  - In section B.1, we demonstrate that the test-time BN adaptation allows the flexibility of appropriately choosing the values of $\sigma$ to certify different test examples using the same AT models.
  - In section B.2, we investigate the effect of test-time BN adaptation for the existing randomized smoothing based models for $\ell_2$ certification (Cohen et al., 2019; Salman et al., 2019a).
  - Table 6 and Table 7 presents the top-1 certified accuracy at different $\ell_2$ radii achieved by different models for ImageNet and CIFAR-10 datasets respectively.

- **Section C: Additional Results under Adversarial Attacks.**
  - In section C.1, we first compare the strength of our adaptive attack as we increase the number of models to approximate the gradients (recall Equation 8).
  - In section C.2, we study the effect of test-time adaptive BN against adversarial attacks when the AT models are over-fitted (i.e., without early stopping criteria during training).

- **Section D: Additional Results on Common Corruptions**
  - In section D.1, we first present the accuracy attained by different classifiers under different common perturbations.
  - In section D.2, we visualize the loss gradients of AT models using adaptive BN technique under different corruptions from ImageNet-C dataset.

Appendix A. Experimental Setup

A.1 Implementation Details

We present our experiments on CIFAR-10 (Krizhevsky et al., 2009) and ImageNet (Deng et al., 2009) datasets. We train the AT models using early stopping criteria as described in (Rice et al., 2020).
A.1.1 CIFAR-10.

We use pre-activation ResNet18 architecture (He et al., 2016b) for our experiments on CIFAR-10. We apply the SGD optimizer with a batch size of 128. We execute a total of 200 training epochs and apply a step-wise learning rate decay set initially at 0.1 and divided by 10 at 100 and 150 epochs, and weight decay $5 \times 10^{-4}$.

**AT models (Madry et al., 2018; Rice et al., 2020):** Unless and otherwise specified, we use the following AT models for our experiments: We learn Adv$_\infty$ and Adv$_2$, for $\ell_\infty$ and $\ell_2$ threat models with threat boundaries of $\epsilon = 8/255$ and $\epsilon = 1$, respectively.

We also learn a few additional AT models with different threat boundaries for our experiments in section 4.3. We denote them by specifying their corresponding threat model and threat boundaries. For example, Adv$_2[\ell_2 \leq 1.5]$ denotes an AT model that is learned using PGD adversary with $\ell_2$ threat model and a threat boundary of $\epsilon = 1.5$, along with early-stopping criteria (Rice et al., 2020).

We also learned AT models without using early-stopping criteria, as in (Madry et al., 2018) for our comparison in Figure 4. These models are denoted as Adv$_{overfit}$.

We use projected gradient descent (PGD) adversarial attack (Madry et al., 2018) to train these AT models as follows: For Adv$_\infty$, we use 10 iterations and an $\ell_\infty$ step size of $\epsilon/4$ (see Equation 7). For Adv$_2$, we use 10 iterations and an $\ell_2$ step size of $\epsilon/8.5$. This is the same experimental setup as in (Rice et al., 2020). We choose a small set of 1,000 images from the CIFAR-10 test set for our validation. We apply the PGD attack with the same hyper-parameters as training for validation. We save the best model using the early-stopping criteria (Rice et al., 2020).

**Randomized smoothing model by Cohen et al. (2019):** We also train Rand$_{\sigma=0.5}$ by training with augmented random noise, sampled from an isotropic Gaussian distribution $\mathcal{N}(0, \sigma^2 I)$ with $\sigma = 0.5$, as in (Cohen et al., 2019). Here, we keep the same model architecture, learning rates, batch sizes, and other hyper-parameters as used to learn the AT models.

**Randomized smoothing model by Salman et al. (2019a):** We also compare with the state-of-the-art certification models, ‘SmoothAdv’ by Salman et al. (2019a) for our experiments on certification using randomized smoothing in section 4.3. We train the SmoothAdv models by choosing random noise vectors followed by an adaptive adversarial attack with specified $\ell_2$ threat boundary of $\epsilon$ at each iteration. The noise vectors are sampled from an isotropic Gaussian distribution $\mathcal{N}(0, \sigma^2 I)$.

We note that the training hyper-parameter $\epsilon$ has the most significant impact on the certification curve for a SmoothAdv model (please refer to Table 7-15 in (Salman et al., 2019a) for more details). For our experiments, we train 4 different SmoothAdv models with $\epsilon = \{0.25, 0.5, 1, 2\}$ and $\sigma = 0.5$ using adaptive PGD attack with 10 steps. We denote them as SmoothAdv$_{\sigma=0.5, \epsilon=0.25}$, SmoothAdv$_{\sigma=0.5, \epsilon=0.5}$, SmoothAdv$_{\sigma=0.5, \epsilon=1}$, and SmoothAdv$_{\sigma=0.5, \epsilon=2}$ respectively. We use the same training set-up and other hyper-parameters as specified in their Github page 4.

---

4. https://github.com/Hadisalman/smoothing-adversarial
A.1.2 ImageNet.

We use ResNet50 architecture (He et al., 2016a) for ImageNet. We obtain the Baseline and Randₜₜ=0.5 models from (Cohen et al., 2019)⁵. These models are trained using Gaussian augmented noises, sampled from isotropic Gaussian distribution \( N(0, \sigma^2 I) \) with \( \sigma = 0.0 \) (i.e., no noise) and \( \sigma = 0.5 \) respectively.

The AT models i.e., Advₜₜ and Adv₂ are learned for \( \ell_\infty \) and \( \ell_2 \) threat models with threat boundary of 4/255 and 3, respectively. We use the publicly available models provided by Rice et al. (2020)⁶. These models are originally adopted from PGD-based adversarial training of the models provided by Engstrom et al. (2019)⁷.

We resize the input images to 256 × 256 pixels and crop 224 × 224 pixels from the center. Note that, ImageNet-C images are already resized and cropped. So, we directly use these images for our experiment in section 4.5. For our experiments on certification, we use a subsample of 500 test images by choosing at most 1 sample for each class (section 4.3). Also, we use a subsample of 2,000 test images, with exactly 2 samples per class, to evaluate the empirical robustness of AT models against adversarial attack (section 4.4).

A.2 Choice of Test-Time Adaptive BN hyper-parameters

Test-time adaptive BN technique is controlled by two hyper-parameters, i.e., the test batch-size and momentum (\( \rho \)) (see Equation 5) to update the statistics of the batch-normalization layers. We assume that the images in the test batches are obtained independently from the same test distribution. We compute the BN statistics corresponding to these images. The hyper-parameter \( \rho \in [0, 1] \) controls the tread-off between precomputed training statistics and test statistics. We obtain a better estimation of the test distribution from a large test batch. Hence, we can choose a higher value of \( \rho \). However, different factors, e.g., available hardware resources, may also constrain the choice of using large test batches in practice.

Here, we compare the top-1 test accuracy under Gaussian augmented noise with \( \sigma = 0.5 \) for different choices of \( \rho \) and the batch size. In Table 1, we note that the standard DNN models (i.e., Baseline) does not improve under Gaussian noise at \( \sigma = 0.5 \) even after BN adaptation. Hence, we skip these models from the following analysis. Also, we refer to the previous work in (Schneider et al., 2020; Nado et al., 2020) that explicitly analyzed the effects of these hyper-parameters for the standard DNN classifiers.

Momentum (\( \rho \)). We first investigate the effect of momentum (\( \rho \)) as we choose a large batch size of 512. In Table 4, we present the performance of AT models for different values of \( \rho \). Recall that, \( \rho = 1 \) denotes ‘full adaptation’ (Equation 5). Here, we completely ignore the training statistics and recompute the BN statistics using the test batch. In contrast, \( \rho = 0 \) represents ‘no adaptation’ similar to the standard ‘deterministic’ inference setup. In this case, we use the precomputed BN statistics obtained during training.

We observe that for ImageNet (Table 4 [Left]) the performance started converging at \( \rho = 0.7 \). For CIFAR-10 (Table 4 [Right]), the convergence started even earlier at \( \rho = 0.5 \).

⁵. https://github.com/locuslab/smoothing
⁶. https://github.com/locuslab/robust_overfitting
⁷. https://github.com/MadryLab/robustness
Appendix B. Additional Results on Certification

B.1 Flexibility of choosing different $\sigma$ for Certification using BN Adaptation

The choice of $\sigma$ controls the tread-off between certified robust accuracy and certification boundary (Cohen et al., 2019; Salman et al., 2019a; Li et al., 2019): When $\sigma$ is small, we can certify smaller radii with high accuracy. However, we cannot certify for large $\ell_2$ radii at all. In contrast, as we choose a higher value for $\sigma$, larger radii can be certified. However, we get a lower certification accuracy at $\ell_2$ radii.

Existing randomized smoothing-based frameworks fix the value of $\sigma$ to choose the random noise during training to learn their base classifier (Cohen et al. 2019; Salman et al., 2019a). They use the same values of $\sigma$ during certification as well. Further, their performance typically degrades as we use a different value of $\sigma$, as shown in Section B.2. In contrast, our...
4.1. Refer to Table 6 for complete results of all models and different settings.

Monte-Carlo sampling with 100 proposed adaptive BN for AT models provides the flexibility to choose different values of optimum adaptation and certification for the same classification model. Adaptive BN allows us to choose those optimum \( \sigma \) for each test example, we can obtain an upper envelope of these curves as our certified top-1 accuracy from a single AT model. \( \text{Adv}_\infty \) and \( \text{Adv}_2 \) models are as defined in section 4.1. Refer to Table 6 for complete results of all models and different settings.

Figure 5: ImageNet: Certified top-1 accuracy at various \( \ell_2 \) radii as we vary \( \sigma \) for BN adaptation and certification for the same classification model. Adaptive BN allows us to select appropriate values of \( \sigma \) to certify different test examples. By selecting those optimum \( \sigma \) for each test example, we can obtain an upper envelope of these curves as our certified top-1 accuracy from a single AT model. \( \text{Adv}_\infty \) and \( \text{Adv}_2 \) models are defined in section 4.1. Refer to Table 6 for complete results of all models and different settings.

Figure 6: CIFAR-10: Certified top-1 accuracy at various \( \ell_2 \) radii as we vary \( \sigma \) for BN adaptation and certification for the same classification model. Adaptive BN allows us to choose different values of optimum \( \sigma \) for certifying different test examples. By selecting those optimum \( \sigma \) for each test example, we can obtain an upper envelope of these curves as our certified top-1 accuracy. Refer to Table 7 for complete results of all models and different settings.

The proposed adaptive BN for AT models provides the flexibility to choose different values of \( \sigma \) separately for each test example during inference while using the same robust classifier.

Figure 5 and Figure 6 demonstrate the results of ImageNet and CIFAR-10 datasets respectively as we choose different values of \( \sigma \) i.e., \( \{0.25, 0.5, 0.75\} \) to adapt the same models during testing. Similar to section 4.3, we estimate the class label probabilities using Monte-Carlo sampling with 100,000 noisy samples and certify the test examples with 99.9%
probability. For ImageNet, we use the same Adv$_\infty$ and Adv$_2$ models as described in Section 4.1. For CIFAR-10, we present the results for different Adv$_\infty$ and Adv$_2$ models, trained with various threat boundaries. We note that we obtain 3 different certification curves for the same AT model as we apply test-time adaptive BN using different values of $\sigma$. Hence, unlike existing randomized smoothing models (Cohen et al., 2019; Salman et al., 2019a), we can get higher upper envelopes of certification curves by choosing the appropriate values of $\sigma$ for different test examples from a single robust model.

However, finding the optimal value of $\sigma$ for certifying different test examples can be expensive. For each example, it would require applying the BN adaptation on the AT model and find the certification boundaries for different values of $\sigma$ to obtain the largest $\ell_2$ certification radius.

![Certification curves](image)

Figure 7: Certified top-1 accuracy at various $\ell_2$ radii as we vary $\sigma$ for BN adaptation and certification for the existing randomized smoothing-based model. (a) Rand$_{\sigma=0.5}$ (Cohen et al., 2019) for ImageNet. (b) Rand$_{\sigma=0.5}$ (Cohen et al., 2019) for CIFAR-10. (c) SmoothAdv (Salman et al., 2019a) for CIFAR-10. Please refer to Table 6 and Table 7 for complete results of all models with different setups for ImageNet and CIFAR-10 datasets.
B.2 Adaptive BN for existing Randomized Smoothing models

Next, we investigate the effect of the test-time adaptive BN technique for the existing randomized smoothing-based certification models (Cohen et al., 2019; Salman et al., 2019a).

In Figure 7, we present the results for ImageNet and CIFAR-10 datasets respectively. As before, we choose three different values of $\sigma$, i.e., $\{0.25, 0.50, 0.75\}$ to adapt the networks during testing phase, that are trained using $\sigma = 0.50$. We observe that the choice of $\sigma = 0.25$ and $\sigma = 0.5$ (i.e., same noise as training) for test-time BN adaptation leads to improve the certification accuracy for Rand$_{\sigma=0.5}$ model on CIFAR-10 dataset. However, in general test-time BN adaptation typically degrades the certification performance of the state-of-the-art SmoothAdv models (Salman et al., 2019a) for both datasets and Rand$_{\sigma=0.5}$ model (Cohen et al., 2019) for ImageNet.

| Model       | BN adaption | Certification | $\ell_2$ Radius |
|-------------|-------------|---------------|-----------------|
| Baseline    | -           | at $\sigma = 0.25$ | 7.8 | 4.8 | 3.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| Adv$_m[\ell_\infty \leq 4/255]$ | at $\sigma = 0.25$ | at $\sigma = 0.25$ | 50.0 | 46.4 | 41.6 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
|            | at $\sigma = 0.50$ | at $\sigma = 0.50$ | 43.6 | 39.4 | 35.8 | 31.4 | 27.6 | 23.4 | 18.2 | 0.0 | 0.0 | 0.0 |
|            | at $\sigma = 0.75$ | at $\sigma = 0.75$ | 31.6 | 26.4 | 22.4 | 18.6 | 16.8 | 14.4 | 11.8 | 9.4 | 7.6 | 5.6 |
| Adv$_m[\ell_2 \leq 3.00]$ | at $\sigma = 0.25$ | at $\sigma = 0.25$ | 53.2 | 50.2 | 46.8 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
|            | at $\sigma = 0.50$ | at $\sigma = 0.50$ | 47.0 | 43.0 | 39.0 | 36.4 | 32.8 | 30.8 | 27.0 | 0.0 | 0.0 | 0.0 |
|            | at $\sigma = 0.75$ | at $\sigma = 0.75$ | 37.8 | 32.2 | 28.4 | 26.0 | 22.4 | 20.2 | 19.0 | 17.4 | 14.2 | 12.0 |
| Rand$_{\sigma=0.5}$ | at $\sigma = 0.50$ | - | 60.8 | 54.4 | 47.8 | 39.0 | 34.2 | 29.0 | 23.8 | 0.0 | 0.0 | 0.0 |
|            | at $\sigma = 0.25$ | at $\sigma = 0.25$ | 59.8 | 53.6 | 46.6 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
|            | at $\sigma = 0.50$ | at $\sigma = 0.50$ | 58.6 | 51.0 | 43.8 | 37.4 | 32.2 | 27.4 | 22.4 | 0.0 | 0.0 | 0.0 |
|            | at $\sigma = 0.75$ | at $\sigma = 0.75$ | 48.6 | 41.6 | 36.6 | 31.2 | 26.2 | 22.4 | 18.6 | 16.8 | 12.8 | 8.6 |

Table 6: ImageNet: Certified top-1 accuracy at various $\ell_2$ radii as we vary $\sigma$ for BN adaptation and certification. We use ResNet50 for ImageNet.

Appendix C. Additional Results under Adversarial Attacks

C.1 Effect of the EoT hyper-parameter ($m$)

We recall that AT models with BN adaptation need to approximate the gradients during back-propagation to effectively find the adversarial examples (see Equation 8). We select a set of $m$ different models, $\{f_{\theta(i)}\}_{i=1}^m$ with different BN parameters for this approximation. Each $f_{\theta(i)}$ is produced using adaptive BN to the original classifier $f_\theta$ using test batches, perturbed with randomly chosen $\delta \in \Delta$ from the threat model. We choose $m = 10$ for both CIFAR-10 and ImageNet datasets for our experiments in Table 2 (Section 4.4).

Here, we compare the strength of the adaptive adversarial attack as we vary the number of models (i.e., $m$) to approximate the gradients. Table 8 and Table 9 present the results for ImageNet and CIFAR-10 respectively. We observe that the performances of the AT models do not significantly change as we vary $m$ to approximate the gradients.

C.2 Adversarial Attack on overfitted models

Rice et al. (2020) demonstrated that AT models suffers from overfitting. It significantly degrades their performance against adversarial attacks. In contrast, by applying early-
| Model          | BN adaption | Certification | $\ell_2$ Radius | $\ell_2$ Radius |
|----------------|--------------|---------------|-----------------|-----------------|
| Baseline       | -            | at $\sigma = 0.25$ | 0.25            | 0.5             |
| Adv$_{\infty}[\ell_\infty \leq 4/255]$ | at $\sigma = 0.25$ | at $\sigma = 0.25$ | 67.96           | 50.46           |
|                | at $\sigma = 0.50$ | at $\sigma = 0.50$ | 47.34           | 31.83           |
|                | at $\sigma = 0.75$ | at $\sigma = 0.75$ | 26.89           | 15.92           |
| Adv$_{\infty}[\ell_\infty \leq 8/255]$ | at $\sigma = 0.25$ | at $\sigma = 0.25$ | 66.43           | 55.06           |
|                | at $\sigma = 0.50$ | at $\sigma = 0.50$ | 53.65           | 42.91           |
|                | at $\sigma = 0.75$ | at $\sigma = 0.75$ | 39.96           | 30.76           |
| Adv$_{\infty}[\ell_\infty \leq 12/255]$ | at $\sigma = 0.25$ | at $\sigma = 0.25$ | 60.52           | 52.42           |
|                | at $\sigma = 0.50$ | at $\sigma = 0.50$ | 51.53           | 43.94           |
|                | at $\sigma = 0.75$ | at $\sigma = 0.75$ | 42.61           | 35.56           |
| Adv$_{\infty}[\ell_\infty \leq 16/255]$ | at $\sigma = 0.25$ | at $\sigma = 0.25$ | 53.75           | 47.57           |
|                | at $\sigma = 0.50$ | at $\sigma = 0.50$ | 48.07           | 42.51           |
|                | at $\sigma = 0.75$ | at $\sigma = 0.75$ | 42.05           | 36.42           |
| Adv$_{\ell_2 \leq 0.50}$ | at $\sigma = 0.25$ | at $\sigma = 0.25$ | 68.84           | 54.04           |
|                | at $\sigma = 0.50$ | at $\sigma = 0.50$ | 48.81           | 33.82           |
|                | at $\sigma = 0.75$ | at $\sigma = 0.75$ | 37.38           | 16.15           |
| Adv$_{\ell_2 \leq 1.00}$ | at $\sigma = 0.25$ | at $\sigma = 0.25$ | 68.02           | 58.54           |
|                | at $\sigma = 0.50$ | at $\sigma = 0.50$ | 56.45           | 46.24           |
|                | at $\sigma = 0.75$ | at $\sigma = 0.75$ | 43.04           | 33.08           |
| Adv$_{\ell_2 \leq 1.25}$ | at $\sigma = 0.25$ | at $\sigma = 0.25$ | 67.13           | 58.77           |
|                | at $\sigma = 0.50$ | at $\sigma = 0.50$ | 57.73           | 48.88           |
|                | at $\sigma = 0.75$ | at $\sigma = 0.75$ | 46.54           | 37.33           |
| Adv$_{\ell_2 \leq 1.50}$ | at $\sigma = 0.25$ | at $\sigma = 0.25$ | 64.21           | 57.13           |
|                | at $\sigma = 0.50$ | at $\sigma = 0.50$ | 56.55           | 49.19           |
|                | at $\sigma = 0.75$ | at $\sigma = 0.75$ | 47.73           | 40.89           |
| Adv$_{\ell_2 \leq 2.00}$ | at $\sigma = 0.25$ | at $\sigma = 0.25$ | 60.4           | 54.71           |
|                | at $\sigma = 0.50$ | at $\sigma = 0.50$ | 54.27           | 48.89           |
|                | at $\sigma = 0.75$ | at $\sigma = 0.75$ | 47.96           | 42.54           |
| Adv$_{\ell_2 \leq 2.25}$ | at $\sigma = 0.25$ | at $\sigma = 0.25$ | 57.08           | 52.5            |
|                | at $\sigma = 0.50$ | at $\sigma = 0.50$ | 52.1           | 46.99           |
|                | at $\sigma = 0.75$ | at $\sigma = 0.75$ | 46.45           | 41.71           |
| Adv$_{\ell_2 \leq 2.50}$ | at $\sigma = 0.25$ | at $\sigma = 0.25$ | 54.88           | 50.79           |
|                | at $\sigma = 0.50$ | at $\sigma = 0.50$ | 50.53           | 46.26           |
|                | at $\sigma = 0.75$ | at $\sigma = 0.75$ | 45.95           | 41.89           |
| Adv$_{\ell_2 \leq 3.00}$ | at $\sigma = 0.25$ | at $\sigma = 0.25$ | 53.82           | 49.69           |
|                | at $\sigma = 0.50$ | at $\sigma = 0.50$ | 49.41           | 45.57           |
|                | at $\sigma = 0.75$ | at $\sigma = 0.75$ | 45.37           | 41.54           |
| Rand$_{\sigma=0.5}$ | -            | at $\sigma = 0.50$ | 51.68           | 40.38           |
|                | at $\sigma = 0.25$ | at $\sigma = 0.25$ | 62.91           | 52.25           |
|                | at $\sigma = 0.50$ | at $\sigma = 0.50$ | 57.58           | 46.46           |
|                | at $\sigma = 0.75$ | at $\sigma = 0.75$ | 46.4           | 35.63           |
| SmoothAdv$_{\sigma=0.5, \ell_2 \leq 0.25}$ | -            | at $\sigma = 0.50$ | 57.8            | 47.63           |
|                | at $\sigma = 0.25$ | at $\sigma = 0.25$ | 58.74           | 48.29           |
|                | at $\sigma = 0.50$ | at $\sigma = 0.50$ | 54.0           | 42.91           |
|                | at $\sigma = 0.75$ | at $\sigma = 0.75$ | 43.15           | 32.29           |
| SmoothAdv$_{\sigma=0.5, \ell_2 \leq 0.50}$ | -            | at $\sigma = 0.50$ | 58.82           | 49.68           |
|                | at $\sigma = 0.25$ | at $\sigma = 0.25$ | 59.89           | 50.4           |
|                | at $\sigma = 0.50$ | at $\sigma = 0.50$ | 55.73           | 45.79           |
|                | at $\sigma = 0.75$ | at $\sigma = 0.75$ | 46.25           | 36.72           |
| SmoothAdv$_{\sigma=0.5, \ell_2 \leq 1.00}$ | -            | at $\sigma = 0.50$ | 56.54           | 49.53           |
|                | at $\sigma = 0.25$ | at $\sigma = 0.25$ | 57.13           | 48.38           |
|                | at $\sigma = 0.50$ | at $\sigma = 0.50$ | 53.56           | 45.79           |
|                | at $\sigma = 0.75$ | at $\sigma = 0.75$ | 47.17           | 39.4           |
| SmoothAdv$_{\sigma=0.5, \ell_2 \leq 2.00}$ | -            | at $\sigma = 0.50$ | 52.82           | 47.67           |
|                | at $\sigma = 0.25$ | at $\sigma = 0.25$ | 52.23           | 47.24           |
|                | at $\sigma = 0.50$ | at $\sigma = 0.50$ | 50.28           | 45.24           |
|                | at $\sigma = 0.75$ | at $\sigma = 0.75$ | 46.7           | 41.79           |

Table 7: CIFAR-10: Certified top-1 accuracy at various $\ell_2$ radii as we vary $\sigma$ for BN adaptation and certification. We use preactivation ResNet18 for CIFAR-10 (Bottom).
Adversarially Trained Models with Test-Time Covariate Shift Adaptation

Table 8: ImageNet: Performance under adversarial attacks as we vary the number of models \((m)\) to approximate the loss gradients for the adversarial attack. For models with BN adaptation, we randomly shuffle the test images to report \((mean + 2 \times sd)\) of 3 different runs.

Table 9: CIFAR-10: Performance under adversarial attacks as we vary the number of models \((m)\) to approximate the loss gradients for the adversarial attack. For models with BN adaptation, we randomly shuffle the test images to report \((mean + 2 \times sd)\) of 3 different runs.

Table 10: Performance under adversarial attacks on CIFAR-10 dataset. Adv\(^{overfit}\) denotes the AT model that is saved after 200 training epochs (as proposed in Madry et al. (2018)). For models with BN adaptation, we randomly shuffle the test images to report \((mean + 2 \times sd)\) of 3 different runs.

stopping criteria during training, we can significantly improve their performance. AT models without early stopping criteria are denoted as Adv\(^{overfit}\).

Here, we study the effect of test-time adaptive BN as we obtain the models with and without using early stopping criteria. In Table 10, we investigate this performance on CIFAR-10. Overall, we observe that test-time adaptive BN produces a slightly higher performance gain for Adv\(^{overfit}\) models. In other words, test-time adaptive BN is more beneficial to improve the empirical robustness for Adv\(^{overfit}\) models.

Appendix D. Additional Results on Common Corruptions

D.1 Performance under different corruptions

Hendrycks and Dietterich (2019) introduced ImageNet-C and CIFAR-10-C datasets by algorithmically generated corruptions from noise, blur, weather, or digital categories with
\(I = 5\) different severities for each corruption. In Table 11 and Table 12, we present the top-1 accuracy of ImageNet and CIFAR-10 classifiers under each corruption, averaged over their 5 severity levels:

\[
\text{Top-1 Acc.}(f)_k = \left(1 - \text{avg. err}(f)_k\right) = \left(1 - \frac{\sum_i \text{err}_{k,i}^f}{I}\right)
\]

(11)

where, \(f\) denotes the classifier, \(k\) is the corruption type and \(i\) is the severity level. \(\text{avg. err}(f)_k\) is the average error for corruption \(k\). \(\text{err}_{k,i}^f\) is the error for corruption \(k\) with severity level \(i\). \(I\) is the total number of severities.

Recall that in Section 4.5, we compute the corruption robustness metrics (i.e., \(mCE\) and \(rmCE\) scores) by normalizing the top-1 error rate for individual corruptions using the corresponding AlexNet error rates. For CIFAR-10-C, we train an AlexNet model (Krizhevsky et al., 2012). We use the SGD optimizer with a constant learning rate of 0.001 for 150 epochs with a batch size of 128 without any data augmentation. For ImageNet-C, Hendrycks and Dietterich (2019) already provided the top-1 error rates for each corruption corresponding to AlexNet model to compute the \(mCE\) and \(rmCE\) scores.

In Table 11 and Table 12, we present the average Top-1 accuracies for each corruption type for ImageNet and CIFAR-10 datasets respectively. Specifically for ImageNet-C, the top-1 accuracy AT models without test-time adaptive BN significantly dropped for corruptions under noise, blur, and weather category, compared to their clean test accuracy. These models often achieve significantly lower accuracy compared to the Baseline model without BN adaptation. For example, \(\text{Adv}_\infty\) and \(\text{Adv}_2\) achieve an average top-1 accuracy of 26.5% and 24.1% respectively, compared to 43.0% for the Baseline model without BN adaptation. In contrast, test-time adaptive BN significantly improves their top-1 accuracy for ImageNet-C.

### D.2 Loss Gradients under different perturbations

In this section, we investigate their loss gradients under random corruptions of different categories. In Section 4.2, we have visualized that under high Gaussian noise, AT models tend to produce noisier loss gradient patterns. However, in Figure 8 we note that the characteristics of the loss-gradients change depending on the types of corruption. As shown by Yin et al. (2019), we can group the random corruptions by analyzing their Fourier spectrums. 8 Towards this, in Figure 8, we visualize the Fourier spectrum of the corruption images along with the loss gradients values for \(\text{Adv}_\infty\) model with and without applying test-time adaptive test-BN. In Figure 9-14, we also visualize the loss gradients corresponding to input image pixels for both \(\text{Adv}_\infty\) and \(\text{Adv}_2\) models using and without using test-time adaptive BN.

In Figure 8, we categorize the corruptions into three groups based on the concentration of Fourier spectrums. We note that corruption images with larger concentrations for lower Fourier frequencies (called “low” frequency corruption) produce blurry loss-gradients for AT models (left column in Figure 8). Hence, the classifier produces almost uniform importance to all input pixels to predict these corrupted images. Typically, we observe this pattern for

---

8. Please refer to (Yin et al., 2019) (see Figure 2) for more details of Fourier perspective on random corruptions.
Table 11: ImageNet-C: Top-1 Accuracy achieved by different classification models under each corruption (averaged over 5 severity levels). For models with BN adaptation, we randomly shuffle the test images to report \((mean + 2 \times sd)\) of 3 different runs.

corruption images from blur and weather categories. Consequently, as the concentrations for higher Fourier frequencies (left to right in Figure 8)), we get sharper loss-gradient patterns.

Finally, we observe that test-time BN adaption mitigates the blurry loss-gradient patterns for corruption images from blur and weather categories as well as the noisy loss-gradients for corruption images of the noise categories. In other words, it mitigates the effect of any random corruption to restore the alignment with perceptually relevant features. It allows the AT models with test-time BN adaption to produce higher top-1 accuracy against any random corruptions (Table 11 and Table 12).
## Table 12: CIFAR-10-C: Top-1 Accuracy achieved by different classification models under each individual corruption (averaged over 5 severity levels). For models with BN adaptation, we randomly shuffle the test images to report \((mean + 2 \times sd)\) of 3 different runs.
Figure 8: Visual comparison of loss-gradients corresponding to “lower” and “higher” frequency corruption images. Adaptive BN significantly mitigates the effect of different corruptions to improve perceptual alignments.
Figure 9: Visualizing the loss-gradients produced by AT models on the test examples that are perturbed using Brightness and Contrast corruptions from weather category.
Figure 10: Visualizing the loss-gradients produced by AT models on the test examples that are perturbed using Snow, Frost and Fog corruptions from Weather category.
Figure 11: Visualizing the loss-gradients produced by AT models on the test examples that are perturbed using Defocus-blur and Glass-blur corruptions from Blur category.
Figure 12: Visualizing the loss-gradients produced by AT models on the test examples that are perturbed using Zoom-blur and Motion-blur corruptions from Blur category.
Figure 13: Visualizing the loss-gradients produced by AT models on the test examples that are perturbed using Elastic transformation, Pixelate and JPEG compression corruptions from Digital category.
Figure 14: Visualizing the loss-gradients produced by AT models on the test examples that are perturbed using Gaussian noise, Shot noise and Impulse noise corruptions from Noise category.