NCIS: Neural Contextual Iterative Smoothing for Purifying Adversarial Perturbations

Sungmin Cha¹, Naeun Ko³, Heewoong Choi², Youngjoon Yoo³, ⁴, and Taesup Moon ²
¹ New York University ² ASRI / INMC / Seoul National University ³ NAVER Cloud ⁴ NAVER AI Lab
sungmin.cha@nyu.edu, naeun.ko@navercorp.com, chw0501@snu.ac.kr, youngjoon.yoo@navercorp.com, tsmoon@snu.ac.kr

Abstract

We propose a novel and effective purification-based adversarial defense method against pre-processor blind white- and black-box attacks, without requiring any adversarial training or retraining of the classification model. Based on the observation of the adversarial noise, we propose a simple iterative Gaussian Smoothing (GS) that smooths out adversarial noise and achieves substantially high robust accuracy. To further improve the method, we propose Neural Contextual Iterative Smoothing (NCIS), which trains a blind-spot network (BSN) in a self-supervised manner to reconstruct the discriminative features of the smoothed original image. From the extensive experiments on the large-scale ImageNet, we show that our method achieves both competitive standard accuracy and state-of-the-art robust accuracy against most strong purifier-blind white- and black-box attacks. Also, we propose a new evaluation benchmark based on commercial image classification APIs, including AWS, Azure, Clarifai, and Google, and demonstrate that users can use our method to increase the adversarial robustness of APIs.

1. Introduction

Despite the great success of deep learning-based image classification, it is well known that the classification models are vulnerable to the adversarial attacks, since the initiative reporting from [18]. Among various attempts to defend against such attacks, adversarial training (AT) [29] is regarded as one of the most robust defense methods. However, AT also possesses several limitations [4] as follows: (a) the computational cost is very expensive [38] particularly for the large-scale datasets like ImageNet [13], (b) it is difficult to guarantee the overall generalization capability as the model needs to be re-trained for every different task and adversary [29], and (c) significantly standard accuracy deteriorations has been reported even for a small perturbation budget for the attack [29] cases.

Regarding the above limitations, the input transformation methods [20, 27, 48], which attempt to remove the adversarial noise in the images before feeding them to classifiers, can be thought of as alternatives for AT since they do not require the re-training of the classifier. While such methods are widely regarded as broken by the strong [20] and sophisticated adaptive attacks, e.g., [2], they have been reconsidered recently due to their practical value, often under the term of purification of adversarial attacks [32, 41, 50].

In this paper, we also focus on the adversarial purification setting. In other words, we assume the adversary may (or may not) have full access to the classifier subject to attack, but has no access to the purification model. Such setting may seem relatively weak compared to other stronger attack scenarios, in which the adversary has additional access to the purification model or its output [3, 42]. However, we argue that such a pre-processor blind attack is what we may encounter the most in practice. For a more concrete argument, consider the situation in which a provider of large-scale image classification API would want to make the classifier “robust” to the potential adversarial attacks [36], while maintaining the standard accuracy as high as possible. In such a case, allowing the strong adaptive attack would be unrealistic since it basically means the entire API service system has been breached by the adversary to access the classifier and purifier, concerning the adversarial robustness of the classifier becomes a secondary issue. A more realistic scenario would be the one where the adversary aims to attack the classifier for the API (with/without the knowledge of the classifier) [28], and API providers devise additional guard, i.e., the purifier for defending the pre-trained classifier.

To this end, we devise a novel smoothing-based purification scheme that can significantly enhance the robustness of the classifier while maintaining the standard accuracy in the above-mentioned setting. Our contributions are threefold. First, we make a novel observation on the distribution of the adversarial noise that it is more or less zero mean and symmetric at the patch level. Second, from the above observation, we show that a very simple Gaussian Smoothing
(GS), which essentially employs a non-negative, symmetric convolution kernel, can achieve surprisingly high robust accuracy when it is iteratively applied as a purifier. Third, in order to compensate for the loss of the standard accuracy of the iterative GS, we employ a novel and efficient Blind-Spot Network (BSN), extended from a recent self-supervised learning-based denoiser [5], to reconstruct the discriminative features in the original image that are smoothed out by the iterative GS.

In order to validate our method, which is dubbed as Neural Contextual Iterative Smoothing (NCIS), we carry out an extensive evaluation of our method on the large-scale ImageNet [13] dataset for the various purifier-blind attack settings. More concretely, we follow the standard experimental protocol proposed in [15] and compare NCIS with other strong recent baselines for the following variations: four different classifier models, (pre-processor blind or full) white-box/black-box attacks, $L_2/L_\infty$ targeted and untargeted attacks, and varying perturbation budget $\epsilon$ and attack iterations. Furthermore, to the best of our knowledge, we propose a new evaluation benchmark for the four commercial vision APIs, i.e., Amazon AWS, Microsoft Azure, Google, and Clarifai, for the first time and evaluate the purifier performance for the strong transfer-based black-box attacks [28] (See Figure 1). Consequently, we demonstrate that our NCIS achieves very robust performance across all the tested attack settings and significantly outperforms the state-of-the-art purifier [32], with 14% fast inference time and $\times 14.7$ lower GPU memory requirement. Moreover, for pre-processor blind white-box $L_\infty$ PGD attack, we show our NCIS even outperforms the strongest adversarial training (AT)-based defense FD [45], without any classifier re-training.

2. Related Work

Blind-spot networks (BSN) for image denoising Recent blind image denoising, training neural network-based denoiser only using noisy images, has made notable improvements. One of the main research directions of denoising is to use the blind-spot network (BSN). Several types of BSN are proposed for their own proposed method [5, 8, 9, 24, 26, 44]. Among them, FBI-Net [5] achieves the best denoising performance with a small-sized network add-on.

Adversarial attack White-box attacks generate adversarial examples based on the input gradient, having network information of target models. FGSM [18] generates adversarial examples with an optimization-based method by a single-step update. In follow-up studies, multi-step methods were proposed which strengthen the level of attacks by taking multiple gradient steps [12, 25, 29] in an iterative manner. Also, other attack approaches including new loss functions [7, 30], momentum-based iterative attack [16], and creating diverse input patterns [46] have been introduced. Black-box attack deals with the case an attacker cannot access the model gradient, and hence more difficult than white-box attack scenario. Transfer-based attacks [28] generate adversarial examples against a substitute model, and it is known that attacks are more successful when the substitute model and the target model architectures are similar [15]. Query-based attacks [1, 10, 22, 43] estimate the gradient through queries but require many queries for adversarial examples generation.

Adversarial defense Adversarial training (AT) methods [18, 29] train the classifier with adversarial examples by min-max optimization, and shows stable robustness against various adversarial attacks. However, they require excessive
training cost and have trade-off [15, 45] between the standard and robust accuracy. Certified defense methods [11, 36] aim to guarantee the model to be robust against adversarial perturbations with theoretical lower bounds on the robust accuracy. Input transformation processes the input images to achieve robustness against adversarial attacks [17, 20, 27, 48]. These methods are simple and cheap to apply but are easily broken by strong adversarial attacks [15]. Recently, purification methods that shift the adversarial examples back to the clean data representation have been actively proposed [7, 16, 18, 25, 29, 46]. We refer to these methods, Denoised Smoothing (DS) [36] proposed on approach that applies a pre-trained Gaussian denoiser for these methods, as simple and cheap to apply but are easily broken by strong adversarial attacks [15]. Recently, purification methods that shift the adversarial examples back to the clean data representation have been actively proposed [7, 16, 18, 25, 29, 46]. We refer to these methods, Denoised Smoothing (DS) [36] proposed an approach that applies a pre-trained Gaussian denoiser for the untargeted attack, where $\mathcal{L}$ for the untargeted attack, where $\mathcal{L}$

denoised noise images in total by generating 5 attacked images, 5,000 adversarial examples with the most generally used, untargeted PGD white-box attack [29]. We collected 5,000 adversarial noise images in total by generating 5 attacked images, 5,000 adversarial examples with the most generally used, untargeted PGD white-box attack [29]. We collected 5,000 adversarial noise images in total by generating 5 attacked images, 5,000 adversarial examples with the most generally used, untargeted PGD white-box attack [29].

Blind-spot network (BSN) The BSN is a special form of neural network that tries to reconstruct a pixel in the middle based on its surrounding context pixels. Namely, for a given input $x$, the output of the BSN for pixel $i$ is denoted by

\[
\hat{y}_i(x) = f(\theta, C_{-i}^{k \times k}),
\]

in which $C_{-i}^{k \times k}$ is the $k \times k$ patch of $x$ surrounding $i$, that does not include the pixel $i$. The BSN is typically used as a denoiser to estimate the clean $x$ based on the noisy $z$, of which the model parameter $\theta$ is trained by minimizing $\| z - f_\theta(z) \|^2_2$, i.e., in a self-supervised manner. In this paper, we utilize BSN as a smoothing function that is trained with the original clean $x$ and show that it can work as an effective purifier that can remove adversarial noise while maintaining the original discriminative feature of $x$. For a particular BSN architecture, we extend the recent state-of-the-art network called FBI-Net in [5].

3.2. Analysis on Adversarial Noise

Suppose we have $n$ clean images $\{x_i\}_{i=1}^n$ and generated adversarial examples $\{x'_i\}_{i=1}^n$ for each clean image by the process (1). Assuming that the adversarial noise is additive, we denote the $i$-th adversarial noise image as

\[
N_i = x'_i - x_i, \quad i = 1, \ldots, n,
\]

and analyze the empirical distribution of $N_i$. More concretely, we randomly selected 1,000 images from the ImageNet training set and generated adversarial examples with the most generally used, untargeted $L_\infty$ PGD white-box attack [29]. We collected 5,000 adversarial noise images in total by generating 5 attacked images for each $x_i$ using the perturbation budgets $\epsilon = \{1/255, 2/255, 4/255, 8/255, 16/255\}$, respectively. Then, we randomly cropped 100 $K \times K$ patches from each noise image (thus, obtained $5 \times 10^5$ patches) and computed the empirical mean and skewness [19] of those patches.

Figure 2(a) shows the empirical mean of the adversarial noise in the patches (for varying patch sizes), and Figure 2(b) shows the empirical skewness. From the figures, we clearly observe that the adversarial noise in a patch is more or less zero-mean and has a symmetric distribution. This somewhat surprising regularity of adversarial noise, although generated from the complex iterative optimization process, motivates us to use a very simple Gaussian Smoothing (GS) based purifier in the next section. Note that additional experimental results for other types of attack are proposed in S.M.

\[
\hat{y} = g_\phi(x).
\]

However, note that the purifier randomly takes the adversarial example $x'$ or the original image $x$ as an input image during the general inference situation, hence it is difficult to be trained in a naive supervised manner.

\[
f_{\theta,i}(x) = f(\theta, C_{-i}^{k \times k}),
\]

where $\hat{y} = g_\phi(x)$. However, note that the purifier randomly takes the adversarial example $x'$ or the original image $x$ as an input image during the general inference situation, hence it is difficult to be trained in a naive supervised manner.

\[
\hat{y} = g_\phi(x).
\]

However, note that the purifier randomly takes the adversarial example $x'$ or the original image $x$ as an input image during the general inference situation, hence it is difficult to be trained in a naive supervised manner.

3. Motivation

3.1. Preliminary and Notations

Adversarial attack In a general image classification task, we denote the original input image by $x \in \mathcal{X} \subset \mathbb{R}^D$, in which $D$ is the number of pixels, and the ground-truth label of $x$ by $y \in \mathcal{Y} \subset \mathbb{R}^L$, in which $L$ is the number of classes. A classification model parameterized by $\phi$ is denoted by $g_\phi : \mathcal{X} \rightarrow \mathcal{Y}$, and we assume $g_\phi$ is pre-trained with external training data. Now, an adversary attempts to attack $g_\phi$ by generating an adversarial example $x'$ for $x$ cause the misclassification of $x$. The attack is called untargeted if $g_\phi(x') \neq y$ and targeted if $g_\phi(x') = y^*$ where $y^*$ is a target class different from $y$. The adversarial example $x'$ is typically generated by solving an optimization problem, e.g.,

\[
x' = \operatorname{argmax}_{z : \| z - x \|_p \leq \epsilon} \mathcal{L}(g_\phi(z), y; \phi),
\]

for the untargeted attack, where $\mathcal{L}$ denotes a classification loss function, $\| z - x \|_p \leq \epsilon$ denotes the $L_p$-norm constraint of the optimization, and $\epsilon$ is the perturbation budget. Directly solving of (1) is intractable and various approximation methods have been proposed [7, 16, 18, 25, 29, 46]. We refer to white-box attack when the model architecture and weight of $g_\phi$ are completely known, and black-box attack vice versa.

Purifier Purification is a pre-processing step for an input image before it is fed to a classification model. Let the goal of the purifier $T_\theta : \mathcal{X} \rightarrow \mathcal{X}$ is to reconstruct the purified image which allows $g_\phi$ to predict an original prediction of $x$ for a given input. Then, given the adversarial example $x'$, the optimal purifier $T_\theta^*$ will be

\[
g_\phi(T_\theta^*(x')) = \hat{y}.
\]
3.3. Iterative Gaussian Smoothing (GS)

Gaussian Smoothing (GS) is a widely used low-pass filter for image processing, which smooths an image with a Gaussian kernel. The mechanism can be represented by the convolution operation between a Gaussian kernel \( k \) of size \( K \times K \) and an input image. Namely, when an adversarial example \( x' \) is given as the input to the Gaussian smoothing function \( G(\cdot) \), we denote the GS process as

\[ G(x') = x' \otimes k, \tag{5} \]

in which \( \otimes \) is the convolution operation. Since \( G(\cdot) \) is a linear operation, and from (4), we have

\[ G(x') = G(x + N') = G(x) + G(N') = G(x) + N' \otimes k \approx G(x). \tag{6} \]

The last approximation follows from \( N' \otimes k \approx 0 \), which is from the observation in Figures 2(a) and 2(b) that the pixels in \( K \times K \) patches in \( N' \) have symmetric, zero mean distribution and \( k \) is a non-negative, symmetric kernel. From (6), therefore, we can deduce that GS can mostly wash out the adversarial noise in \( x' \) and will make the smoothed version of \( x' \) become almost identical to that of \( x \).

Furthermore, by denoting \( G^i = G \circ G^{i-1} \) as applying GS \( i \) times iteratively, we can apply the similar logic as in equation (6) and deduce \( G^i(x') \approx G^i(x) \) as well, but with smaller approximation error. The reason is that the patches in \( N_i \otimes k \) will still have zero mean, symmetric distributions, and the adversarial noise will keep getting washed out as we iteratively apply the Gaussian kernel \( k \). On the other hand, as GS continues, we can also expect that \( G^i(x') \) and \( G^i(x) \) will get farther from \( x \), since GS will also wash out detailed and discriminative features in \( x \). Thus, as the iteration continues, we can conjecture that iterative GS may encounter a trade-off of increasing the robust accuracy, by removing adversarial noise, while hurting the standard accuracy, by also removing the discriminative features in \( x \).

Figure 2(c) and 2(d) experimentally validate the above conjecture for the iterative GS. In the figures, for an ImageNet pre-trained ResNet-152 classifier, we tested with 1,000 clean images, \( x \), randomly subsampled from the ImageNet training dataset, and their adversarial examples, \( x' \), generated by \( L_\infty \) PGD (\( \epsilon = 16/255 \)) attack. Figure 2(c) shows the MSE of \( \| x - G^i(x) \|_2^2 \) and \( \| x - G^i(x') \|_2^2 \) as \( i \) increases, for different patch size \( K \). From the figure, we can clearly observe that the two MSEs become almost identical, but increase (i.e., both \( G^i(x) \) and \( G^i(x') \) get farther from \( x \)), as \( i \) increases, corroborating our above conjecture. Figure 2(d) reports both standard and robust accuracy of ResNet-152 when the iterative GS is used as a purifier for the same \( L_\infty \) PGD attack on the whole ImageNet validation set. We observe that the very simple iterative GS is surprisingly effective in purifying the adversarial noise as the robust accuracy of 44.92% is achieved when \( K = 5 \) and \( i = 7 \). However, we also note that the standard accuracy decreases as \( i \) gets larger due to the removed discriminative features. Motivated by this result, we propose our NCIS, which utilizes the iterative GS to maintain high robust accuracy but also employs BSN that can reconstruct the discriminative features washed out by GS to also achieve high standard accuracy.

4. Neural Contextual Iterative Smoothing

Motivated by the strong performance of iterative GS in the previous section, we propose a learnable neural network-based smoothing function as an iterative smoothing-based purifier. More concretely, we first present the improved performance of using self-supervised trained FBI-Net \([5]\) for iterative smoothing, devise a more efficient version of FBI-Net dubbed as FBI-E, and present our NCIS, combining FBI-E with GS for a more stable and superior purifier.

Iterative smoothing with FBI-Net FBI-Net is a fully convolutional network that utilizes a special class of masked convolution filters as shown in Figure 3(a) such that the BSN condition (3) can be satisfied. Now, for given \( n \) clean images \( \{ x_i \}_{i=1}^n \), we can train the network parameters, \( \theta \), of the FBI-Net by minimizing

\[ \sum_{i=1}^n \| x_i - f_\theta(x_i) \|_2^2, \tag{7} \]
namely, in a self-supervised manner. Denoting the learned parameter by \( \theta^* \), we expect that \( f_{\theta^*}(x) \) can be an effective smoothing-based purifier. The reasoning is, since the adversarial noise in \( x' \) is very small as shown in Figure 2(a), and it is generated independent of \( f_{\theta^*} \), the trained FBI-Net would have the similar outputs for both \( x \) and \( x' \), and they will be a smooth reconstruction of \( x \). Consequently, we expect

\[
\|f_{\theta^*}(x') - x\|^2 \approx \|f_{\theta^*}(x) - x\|^2 < \|G(x') - x\|^2 \approx \|G(x) - x\|^2. \tag{8}
\]

Note the reasoning for (8) is possible since we are using BSN; when an ordinary fully convolutional network-based denoiser, e.g., DnCNN [52], is used for \( f_{\theta} \) in (7), then it will end up learning an identity map, hence, \( \|f_{\theta^*}(x') - x'\|^2 \approx 0 \). Thus, the adversarial example will be preserved.

Figure 5(a), which is for the same setting as Figure 2(c), experimentally verifies the above intuition for the iterative smoothing with the FBI-Net. Namely, denoting \( f_{\theta}^{(i)} = f_{\theta^*} \circ f_{\theta^*}^{-1}(x) \), the figure shows \( \|x - f_{\theta}^{(i)}(x)\|^2 \) and \( \|x - f_{\theta^*}(x')\|^2 \) as the iteration number \( i \) increases (the green solid and dashed line, respectively). Compared to the best iterative GS with \( K = 5 \) (brown solid and dashed lines), we observe that FBI-Net-based smoothing does a much better job than GS in reconstructing the original image \( x \) for both input cases (\( x \) and \( x' \)), until \( i \leq 4 \). Moreover, in Figure 5(b), we show the Purification Success Rate (PSR) as

\[
\text{PSR}(x, z) = \frac{1}{n} \sum_{i=1}^{n} \mathbb{I}\{g_{\theta}(x) = g_{\theta}(z)\}.
\]

in which \( z \) is the output of a purifier that takes either \( x \) or \( x' \) as input. The PSR is a metric for how much the PSR can recover the original classification result for \( x \). From the figure, we again observe that the iterative smoothing with FBI-Net does a much better job than iterative GS, in line with Figure 5(a), in preserving the original classification for \( x \) (green solid), until iteration \( i \leq 6 \). We also observe that PSR for \( x' \) (green dashed) increases well until \( i \leq 6 \).

**FBI-E: Extending FBI-Net** While promising, we also observe that iterative smoothing with FBI-Net shows a few limitations. First, PSR for the given \( x' \) is still lower than iterative GS. Second, MSE and PSR diverge significantly, causing unstable results after a certain iteration. Finally, FBI-Net requires a large memory cost for the reconstruction and has a slow inference time to be used as a purifier.

To overcome these limitations and improve efficiency, we introduce two tensor operations for FBI-Net, \( \text{patch} \rightarrow \text{channel} \) and \( \text{channel} \rightarrow \text{patch} \), to expand the reconstruction process from \( \text{context} \rightarrow \text{pixel} \) to \( \text{context} \rightarrow \text{patch} \). Hence, the FBI-E mapping, \( F_{\theta}(x) \), can be denoted as

\[
F_{\theta}(x) = \mathcal{O}_{P \rightarrow C}(f_{\theta}(\mathcal{O}_{P \rightarrow C}(x))), \tag{9}
\]

in which \( \mathcal{O}_{P \rightarrow C}(\cdot) \) and \( \mathcal{O}_{C \rightarrow P}(\cdot) \) denotes \( \text{patch} \rightarrow \text{channel} \) and \( \text{channel} \rightarrow \text{patch} \) operation, respectively. Figure 3(b) illustrates the two operations being applied as a pre- and post-operation for FBI-Net; \( \mathcal{O}_{P \rightarrow C}(\cdot) \) transfers pixels of each \( m \times m \) patch in an input image (color-coded) to channel-wise pixels and \( \mathcal{O}_{C \rightarrow P}(\cdot) \) exactly does the reverse operation. By these operations, FBI-E is also a BSN, but the reconstruction is now done in a patch level based on the context around the patch, and we expect (8) would also hold for \( F_{\theta} \).

The advantage of the two proposed operations is quite huge since it reduces the spatial resolution of both the input image and all feature maps of FBI-Net by \( m \times m \) times. As shown in Figure 5(a) and 5(b), we experimentally checked that FBI-E with \( m = 2 \) achieves superior MSE and PSR for both input cases (orange and dashed lines) than GS and FBI-Net. Furthermore, the inference time and memory improvement of FBI-E over FBI-Net is given in the ablation study given in the later section.

**Neural contextual iterative smoothing (NCIS)** After applying the proposed two operations, the level of difficulty for the reconstruction increases since FBI-E has to reconstruct the entire patch of size \( m \times m \), not just a single pixel. To address this problem, we propose to combine FBI-Net with GS and propose a new smoothing function as

\[
F_{\theta, \text{NCIS}}(x) = F_{\theta}(x) + G(x). \tag{10}
\]

The network parameter \( \theta^* \) is obtained by minimizing (7), which results in \( F_{\theta}(x) \), learning to reconstruct the residual \( x - G(x) \). The intuition is that, when an adversarial example \( x' \) is given as input, we expect \( G(x') \) to wash out the adversarial noise such that \( G(x') \approx G(x) \), and then we let \( F_{\theta^*}(x') \) to reconstruct the discriminative features of the original \( x \) that is also smoothed out by \( G(x) \). Hence, we expect \( F_{\theta^*}(x') \approx F_{\theta}(x) \approx x - G(x) \). Finally, our proposed Neural Contextual Iterative Smoothing (NCIS) is obtained by iteratively applying \( F_{\theta^*}(\text{NCIS}) \) denoted as \( F_{\theta^*}(\text{NCIS}) = F_{\theta^*}(\text{NCIS}) \circ F_{\theta^*}^{-1}(\text{NCIS}) \). The overall procedure of our NCIS is depicted in Figure 4.

In Figure 5(a) and 5(b), we verify the promising results of our NCIS. First, NCIS (\( m = 2 \) for FBI-E and \( K = 11 \) for GS) achieves the lowest MSE for both input cases until the \( i = 7 \) compared to other methods, which shows that
Purification Success Rate

We selected four ImageNet pre-trained models, ResNet-101 [51], ResNeXT-101 [47], and WideResNet-101 [21], for using adversarial attack. We add the DnCNN + GS (MIFGSM [16], DIFGSM [46] and AutoAttack [12]. For gradient-based iterative attacks, PGD [29], CW [7], TVM [p = 0.3, λ = 0.5, max iteration = 10] [20, 35] and SR (σ = 0.04) [31]. Moreover, we implemented the current state-of-the-art purification-based method, NRP and NRP(resG) [32], from their official code. NRP (resG) is the lightweight version of NRP. Lastly, we added FD [45], one of the current state-of-the-art adversarial training method, by implementing the code and weight for ResNet-152 proposed in [15]. Additionally, we had considered adding recently proposed purification methods such as SOAP [40] and ADP [50], but neither of them publicized their code nor experimented on ImageNet. A more detailed comparison of settings with other purifier-based methods is proposed in the S.M. For Gaussian smoothing (GS) used in all experiments including GS in NCIS, we set σ = (K − 1)/6 by following the fact that the length for 99 percentile of Gaussian distribution is 6σ. If there are no additional notations, we set K for GS to 5 and only used NCIS consisting of FBI-E (m = 2) and GS (K = 11) for all experiments. We conducted experiments using NCIS trained by three different seeds and report the average result. The detailed description of the experimental settings, the architecture of FBI-Net, and hyperparameters are in the S.M.

5. Experimental Results

5.1. Experimental settings

In this section, we validate our NCIS against various types of attacks on the ImageNet validation dataset. We designed our experimental setting following [6, 15] for rigorous verification of our proposed method. Also, we used attacks and defenses implemented on public packages [14, 15, 23, 33].

Adversarial attack For using white-box attack methods, we mainly consider pre-processor blind white-box attack, which has access only to classification model weights. We selected four ImageNet pre-trained models, ResNet-152 [21], WideResNet-101 [51], ResNeXT-101 [47] and RegNet-32G [34], and then we generated adversarial examples of the ImageNet validation dataset using four untargeted gradient-based iterative attacks, PGD [29], CW [7], MIFGSM [16], DIFGSM [46] and AutoAttack [12]. For black-box attack, we experimented with transfer-based attacks which still achieve a strong and efficient attack success rate compared to other types of black-box attacks. For transfer-based attack, we generated adversarial examples by attacking a substitute model with \( L_\infty \) PGD attack, and the substitute models are listed in the S.M.

Adversarial defense We selected four input transformation-based defense methods abbreviated as FS (depth = 4) [49], JPEG (quality = 90) [17], TVM (p = 0.3, λ = 0.5, max iteration = 10) [20, 35] and SR (σ = 0.04) [31]. Moreover, we implemented the current state-of-the-art purification-based method, NRP and NRP(resG) [32], from their official code. NRP (resG) is the lightweight version of NRP. Lastly, we added FD [45], one of the current state-of-the-art adversarial training method, by implementing the code and weight for ResNet-152 proposed in [15]. Additionally, we had considered adding recently proposed purification methods such as SOAP [40] and ADP [50], but neither of them publicized their code nor experimented on ImageNet. A more detailed comparison of settings with other purifier-based methods is proposed in the S.M. For Gaussian smoothing (GS) used in all experiments including GS in NCIS, we set σ = (K − 1)/6 by following the fact that the length for 99 percentile of Gaussian distribution is 6σ. If there are no additional notations, we set K for GS to 5 and only used NCIS consisting of FBI-E (m = 2) and GS (K = 11) for all experiments. We conducted experiments using NCIS trained by three different seeds and report the average result. The detailed description of the experimental settings, the architecture of FBI-Net, and hyperparameters are in the S.M.

5.2. Experimental results for pre-processor blind white-box attacks

Experiments with various white-box attacks Table 1 shows the experimental results of each defense method against five attacks for four different ImageNet pre-trained classification models. First, as for standard accuracy, JPEG achieved the best results. However, as already presented in [15, 20], traditional input transformation-based methods are easily broken by white-box attacks. Second, GS achieves superior robust accuracy in most cases and even outperforms NRP and NRP (resG). One possible reason NRP and NRP (resG) show deteriorated performance compared to its original paper is that the proposed loss function for training NRP and NRP (resG) is not well generalized to purify adversarial
Table 1. Experimental results of untargeted white-box adversarial attack. For $L_\infty$ attacks, we set $\epsilon = 16/255$, $\alpha = 1/255$ and attack iterations = 10. For $L_2$ PGD attack, we used $\epsilon = 5$ and $\alpha = 0.1$. For $L_2$ CW attack, other than setting attack iterations as 10, we applied default hyperparameters proposed in [23]. Boldface denotes our proposed methods, and red and blue denotes the highest and second highest results respectively. For ResNet-152, WideResNet-101 and ResNeXt-101, we set $i = 7$ for GS and NCIS.

<table>
<thead>
<tr>
<th>Model / Defense</th>
<th>Standard Accuracy</th>
<th>CW ((L_\infty))</th>
<th>MIFGSM ((L_\infty))</th>
<th>DIFGSM ((L_\infty))</th>
<th>PGD ((L_\infty))</th>
<th>PGD ((L_2))</th>
</tr>
</thead>
<tbody>
<tr>
<td>ResNet-152</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>W/o defense</td>
<td>85.24</td>
<td>97.27</td>
<td>6.34</td>
<td>0.43</td>
<td>6.20</td>
<td>10.69</td>
</tr>
<tr>
<td>JPEG</td>
<td>78.13</td>
<td>76.78</td>
<td>6.36</td>
<td>0.61</td>
<td>6.25</td>
<td>29.01</td>
</tr>
<tr>
<td>FS</td>
<td>76.66</td>
<td>46.37</td>
<td>6.35</td>
<td>0.46</td>
<td>6.22</td>
<td>41.08</td>
</tr>
<tr>
<td>TVM</td>
<td>69.84</td>
<td>59.37</td>
<td>9.32</td>
<td>5.02</td>
<td>17.41</td>
<td>59.15</td>
</tr>
<tr>
<td>SR</td>
<td>77.24</td>
<td>40.87</td>
<td>6.36</td>
<td>0.05</td>
<td>6.22</td>
<td>31.06</td>
</tr>
<tr>
<td>NRP (resG)</td>
<td>72.23</td>
<td>58.05</td>
<td>16.39</td>
<td>2.40</td>
<td>9.60</td>
<td>55.84</td>
</tr>
<tr>
<td>NRP</td>
<td>74.04</td>
<td>58.16</td>
<td>12.35</td>
<td>2.59</td>
<td>10.71</td>
<td>55.58</td>
</tr>
<tr>
<td>GS</td>
<td>68.93</td>
<td>60.36</td>
<td>24.28</td>
<td>22.28</td>
<td>44.92</td>
<td>60.65</td>
</tr>
<tr>
<td>NCIS</td>
<td>68.93</td>
<td>64.32</td>
<td>39.05</td>
<td>33.29</td>
<td>48.06</td>
<td>64.28</td>
</tr>
</tbody>
</table>

| WideResNet-101  |                   |                    |                        |                        |                   |                 |
| W/o defense     | 78.91             | 92.42              | 6.96                   | 0.31                   | 6.60              | 11.97           |
| JPEG            | 78.22             | 41.55              | 6.95                   | 0.10                   | 6.91              | 33.19           |
| FS              | 77.03             | 48.82              | 6.95                   | 0.10                   | 6.91              | 46.41           |
| TVM             | 69.17             | 69.18              | 12.15                  | 7.56                   | 20.33             | 59.84           |
| SR              | 77.85             | 40.24              | 6.94                   | 0.10                   | 6.93              | 33.14           |
| NRP (resG)      | 72.43             | 58.96              | 19.66                  | 4.34                   | 12.18             | 58.17           |
| NRP             | 74.58             | 58.60              | 14.97                  | 5.53                   | 13.22             | 57.69           |
| GS              | 68.54             | 63.96              | 34.18                  | 33.98                  | 49.26             | 64.03           |
| NCIS            | 68.54             | 63.96              | 34.18                  | 33.98                  | 49.26             | 64.03           |

| ResNeXt-101     |                   |                    |                        |                        |                   |                 |
| W/o defense     | 79.21             | 94.33              | 8.59                   | 0.61                   | 7.81              | 13.84           |
| JPEG            | 78.28             | 43.23              | 8.46                   | 0.65                   | 7.96              | 36.77           |
| FS              | 77.54             | 47.88              | 8.61                   | 0.62                   | 7.94              | 46.14           |
| TVM             | 70.66             | 59.47              | 12.52                  | 6.68                   | 20.67             | 61.05           |
| SR              | 78.08             | 40.45              | 8.59                   | 0.80                   | 8.0               | 34.31           |
| NRP (resG)      | 73.65             | 50.04              | 20.57                  | 4.14                   | 13.60             | 59.53           |
| NRP             | 75.28             | 58.30              | 15.88                  | 5.23                   | 13.58             | 58.15           |
| GS              | 67.78             | 60.77              | 36.47                  | 35.53                  | 51.46             | 65.57           |
| NCIS            | 70.08             | 65.12              | 36.47                  | 35.53                  | 51.46             | 65.57           |

In the S.M, we present additional results as follows: First, we report experimental results against AutoAttack [12], which is one of the most powerful attack methods, and the results are consistent with our previous findings. Second, in order to compare our method with Denoised Smoothing (DS) [36], we implemented both their official code and the pre-trained weights of a denoiser. We experimentally confirm that not only DS requires a significant amount of time for defense, but its robust accuracy against PGD attack is also lower than that of our method. Third, we compare the inference time, GPU memory usage, and number of parameters of each method. As a result, we confirm that NRP has slower inference time than NCIS due to its larger number of parameters and the increased GPU memory usage required for purification.

Experiments with PGD attack with various settings To evaluate the proposed method more rigorously, we experimented with both targeted and untargeted $L_\infty$ PGD attack with various $\epsilon$ and the number of attack iterations.

Figure 6(a) and 6(b) show the experimental results for untargeted $L_\infty$ PGD attack in ResNet-152 [21]. As $\epsilon$ and the number of attack iterations increase, the robust accuracy of baseline defense methods goes significantly down. Besides, FD has lower standard accuracy (63.96) compared to NCIS. Moreover, input transformation-based methods and NRP are crumbling like a wreck when the attack setting is strong. However, NCIS and GS achieve the highest robust accuracy, showing up to four times higher robust accuracy of FD. Besides, both the standard and robust accuracy of NCIS are higher than those of GS in all settings.

Also, Figure 6(c) and 6(d) show experiments on targeted $L_\infty$ PGD attack. We observe that the tendency of experimental results has changed from the untargeted results. First, FD and NRP (resG) show better performance than the untargeted attack case. We presume that both FD and NRP (resG) have a generalization problem and fail to defend against all types of attacks stably. Second, the robust accuracy of FD and NRP degrade as the number of iterations and $\epsilon$ increase, resulting in lower robust accuracy than our methods against strong
attacks. On top of this, NCIS and GS achieve superior robust accuracy against strong attacks. See S.M for the results on other classifiers (e.g., WideResNet-101, ResNeXt-101, and RegNet-32G), showing similar tendencies.

Although our paper focuses on pre-processor blind white-box attack for a classification model, we also conducted experiments on full and purifier-aware white-box attacks where attackers can access or are aware of the purifier. To counter both strong attacks, we propose a simple variant of NCIS, adding Gaussian noise in the procedure of iterative smoothing. Also, note that we conducted ablation studies to confirm the role of each proposed component. The details and experimental results are proposed in the S.M.

### 5.3. Experimental results for black-box attacks

**Transfer-based black-box attack** Figure 7 shows ResNet-152 results attacked by adversarial examples generated by attacking WideResNet-101 [51]. The notable discovery is that FD shows the almost constant but most robust result for all cases, similar to experimental results already proposed in [15]. Also, different from white-box attack cases, NRP (resG) achieves competitive performance compared to other input transformation-based methods, supporting the result of their paper. However, note that our GS and NCIS not only surpass the performance of NRP (resG) but also NCIS achieves competitive performance compared to FD when considering both standard and robust accuracy. Additionally, we conducted experiments with the state-of-the-art score-based black-box attack (Square [1]) and the experimental results on other classifiers, reported in the S.M.

**Experiment with APIs** [28] showed existing APIs for multi-label classification can be fooled by ensemble transfer-based black-box attacks. Based on this finding, we propose a new experiment for evaluating purification methods using APIs of Azure, AWS, Clarifai, and Google. From this experiment, we evaluate how well each purifier can restore the top five labels predicted from a clean image when given strong adversarial examples. To evaluate each purifier, we used three evaluation metrics: Prediction Accuracy, Top-1 Accuracy, and Top-5 Accuracy. The additional details of metrics and used hyperparameters are noted in the S.M.

Table 2 shows experimental results on the generated test dataset. First, we observe that traditional input transformation-based methods generally show high standard accuracy with competitive robust accuracy compared to NRP (resG). We believe that this result shows another example of insufficient generalization of NRP (resG) for purifying various types of adversarial examples. Second, NCIS and GS show the most uniformly competitive performance for various APIs. Among them, NCIS achieves better purification performance than GS, considering the average of the standard and robust accuracy.

### 6. Concluding Remarks

We proposed Neural Contextual Iterative Smoothing (NCIS) for adversarial purification against adversarial attack to a classifier. First, we proposed the novel observation that adversarial noise has almost zero mean and a symmetric distribution. Second, based on the above finding, we proposed the learnable neural network-based smoothing function, named as NCIS, for adversarial purification. From the extensive experiments, we observed NCIS robustly purifies adversarial examples generated from various types of white- and black-box attack without requiring re-training of the classification model.

### 7. Acknowledgment

This work was done while S. Cha did a research internship at NAVER AI Lab. All authors thank NSML team for the GPU support. The work was also supported in part by NRF grant [2021R1A2C2007884, 2021M3E5D2A01024795], IITP grant [No.2021-0-01343, No.2021-0-02068, No.2022-0-00113, No.2022-0-00959] funded by the Korean government, and SNU-Naver Hyperscale AI Center.
References


