Preemptive Image Robustification for Protecting Users against Man-in-the-Middle Adversarial Attacks

Seungyong Moon[*1], **Gaon An**[*1], **Hyun Oh Song**[†12]

[1] Department of Computer Science and Engineering, Seoul National University, Seoul, Korea
[2] DeepMetrics, Seoul, Korea
{symoon11, white0234, hyunoh}@mllab.snu.ac.kr

Abstract

Deep neural networks have become the driving force of modern image recognition systems. However, the vulnerability of neural networks against adversarial attacks poses a serious threat to the people affected by these systems. In this paper, we focus on a real-world threat model where a Man-in-the-Middle adversary maliciously intercepts and perturbs images web users upload online. This type of attack can raise severe ethical concerns on top of simple performance degradation. To prevent this attack, we devise a novel bi-level optimization algorithm that finds points in the vicinity of natural images that are robust to adversarial perturbations. Experiments on CIFAR-10 and ImageNet show our method can effectively robustify natural images within the given modification budget. We also show the proposed method can improve robustness when jointly used with randomized smoothing.

Introduction

Recent progress in deep neural networks has enabled substantial performance gains in various computer vision tasks, including image classification, object detection, and semantic segmentation. Leveraging this advance, more practitioners are deploying neural network-based image recognition systems in real-world applications, such as image tagging or face recognition. However, neural networks are vulnerable to *adversarial examples* (Szegedy et al. 2013), minute input perturbations intentionally designed to mislead networks to yield incorrect predictions. These adversarial examples can significantly degrade the performance of the network models, raising security concerns about their deployment.

When these image recognition systems are deployed to applications where users freely upload images from local machines to remote storage, such as social media, this vulnerability can pose another serious threat, especially to the individual application users. Consider there exists a man-in-the-middle (MitM) adversary that can intercept and add perturbations to the images web users upload during transmission (Figure 1). Then, this adversary can easily vandalize neural network-based web services such as image auto-tagging in social media apps by perturbing the images to be misclassified. This type of attack can severely deteriorate user experience, especially as the adversary can further use this attack to insult the uploaders beyond simple misclassification. Even though the MitM attack is one of the most lethal cyber threats (Desmedt 2011; Li et al. 2019; Wang et al. 2019), protecting neural networks from this type of attack has been much less studied in adversarial machine learning literature.

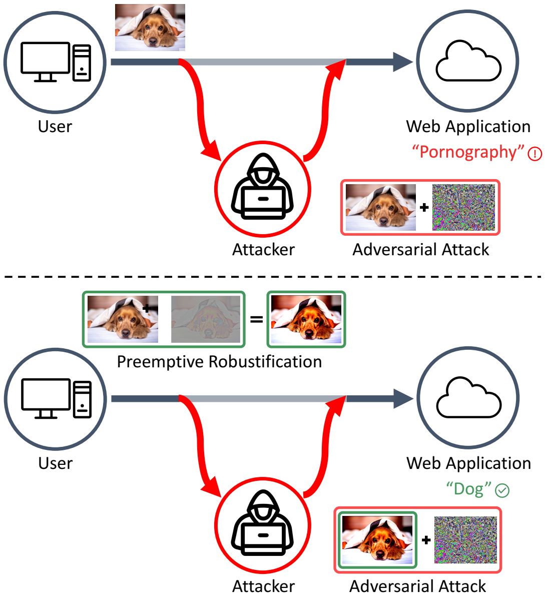

Figure 1: Illustration of our proposed method. Without protection, a MitM adversary can easily perturb the user's image to be misclassified by the web application (top). Our proposed method preemptively robustifies the user's image, securing the image from adversarial attack (bottom).

In this work, we develop a new defense framework to protect web users from MitM attacks. To provide more effective protection measures for the users, we focus on the fact that users hold control of their images before the adversary, unlike in conventional adversarial attack scenarios. Based on this observation, we ask the following question:

- *Can we preemptively manipulate images slightly to be robust against MitM adversarial attacks?*

[*] These authors contributed equally.
[†] Corresponding author

To answer this, we explore the existence of points in image space that are resistant to adversarial perturbations, given a trained classifier. We propose a novel bi-level optimization algorithm for finding those robust points under a given modification budget starting from natural images and measure the degree of robustness achievable by utilizing these points, which we denote as *preemptive robustness*. Moreover, we propose a new network training scheme that further improves preemptive robustness. We validate the effectiveness of our proposed framework in the image classification task on CIFAR-10 (Krizhevsky and Hinton 2009) and ImageNet (Russakovsky et al. 2015) datasets. Our extensive experiments demonstrate that our framework can successfully robustify images against a wide range of adversarial attacks in both black-box and white-box settings. We also observe that our method can enhance the preemptive robustness on smoothed classifiers.

In summary, our main contributions are as follows:

- We introduce a novel real-world adversarial attack scenario targeting users in image recognition systems.
- We propose a new defense framework to improve preemptive robustness and formulate the preemptive robustification process as a bi-level optimization problem.
- We demonstrate our proposed framework can significantly improve preemptive robustness against a wide range of adversarial attacks on standard benchmarks.

Related Works

Adversarial robustness of neural networks Most prior work on adversarial robustness aims to train neural networks that achieve high accuracies on adversarially perturbed inputs. PGD adversarial training improves the empirical robustness of neural networks by augmenting training data with multi-step PGD adversarial examples (Madry et al. 2017). Some recent works report performance gains over PGD adversarial training by modifying the adversarial example generation procedure (Zhang and Wang 2019; Zhang et al. 2020). However, most of the recent algorithmic improvements can be matched by simply using early stopping with PGD adversarial training (Rice, Wong, and Kolter 2020; Croce and Hein 2020). Despite its effectiveness, a major drawback of adversarial training is that it takes a huge computational cost to generate adversarial examples during training. To address this, several works develop fast adversarial training methods by reusing the gradient computation or reducing the number of attack iterations (Shafahi et al. 2019; Zhang et al. 2019a; Wong, Rice, and Kolter 2020).

Although such adversarial training methods can significantly improve the empirical robustness of neural networks, there is no guarantee that a stronger, newly-discovered attack would not break them. To address this, a separate line of work focuses on certifying robustness against any adversarial perturbations (Raghunathan, Steinhardt, and Liang 2018; Wong and Kolter 2018) but often has difficulty in scaling to large neural networks. Randomized smoothing, a method that injects random additive noises to inputs to construct smoothed classifiers from base classifiers, has been considered the most successful certified defense approach that can be applied to large neural networks (Lecuyer et al. 2019; Cohen, Rosenfeld, and Kolter 2019; Salman et al. 2019; Yang et al. 2020).

Preemptive image manipulation for robustness There have been a few studies on preemptive image manipulation to protect images from being exploited, yet most of them utilize it for privacy protection. In the facial recognition task, prior work proposes an algorithm that slightly modifies personal photos before uploading them to social media to make them hard to be identified by malicious person recognition systems (Oh, Fritz, and Schiele 2017; Shan et al. 2020; Cherepanova et al. 2021). Our work differs from this prior work in that our goal is to robustify images to be correctly identified by classification systems even under adversarial perturbations.

The most relevant to our work is an approach of Salman et al. (2020), which develops patches that can boost robustness to common corruptions when applied to clean images. However, we consider the problem of manipulating images to be correctly classified against worst-case adversarial perturbations, which are artificially designed to cause misclassification. Ensuring robustness to adversarial perturbations is much more challenging than robustifying images against common corruptions, which are not the worst-case perturbations.

Methods

Problem Setup

To start, we introduce our defense framework for image classification models, along with the adversarial threat model. In our framework, a defender can preemptively modify the original image x_o to produce a new image x_r that is visually indistinguishable from x_o ahead of adversarial attacks. After the modification, the defender discards the original image x_o, so that adversary can only see the modified image x_r. Under this framework, we can consider two types of adversaries:

- A *grey-box* adversary who has complete knowledge of the classification model but does not know the modification algorithm exists.
- A *white-box* adversary who not only has full access to the classification model but also is aware of the existence of the modification algorithm and how it works.

The grey-box adversary will regard the given image x_r as the original image and attempt to find an adversarial example near x_r, as from the conventional adversarial literature. In contrast, the white-box adversary recognizes that x_r is a modified version. Thus, the white-box adversary will instead try to guess the location of the original image x_o and craft an adversarial example near it.

In this paper, we investigate the defender's optimal strategy for manipulating original images to be resistant against these two adversaries. First, we develop an algorithm that preemptively robustifies original images against the grey-box adversary. Then, we demonstrate our proposed algorithm also exhibits high robustness against adaptively designed white-box attacks.

Preemptive Robustness

We now formally introduce the concept of *preemptive robustness*. We begin by recalling the definition of adversarial examples. Let $c : \mathcal{X} \to \mathcal{Y}$ be a classifier which maps images to class labels. Given an original image $x_o \in \mathcal{X}$ and its class $y_o \in \mathcal{Y}$, suppose x_o is correctly classified. Then, an adversarial example $x_o^a \in \mathcal{X}$ of x_o is defined as an image in the neighborhood of x_o such that the classifier changes its prediction, *i.e.*, $c(x_o^a) \neq c(x_o)$ and $x_o^a \in B_\epsilon(x_o)$. Here, $\epsilon > 0$ is the perturbation budget of the adversary and $B_\epsilon(x) = \{x' \in \mathcal{X} \mid \|x' - x\|_p \leq \epsilon\}$ denotes the closed ℓ_p-ball of radius ϵ centered at x. Throughout this paper, we consider $p \in \{2, \infty\}$, the most common settings in adversarial machine learning literature. If the classifier gives robust predictions in the neighborhood of x_o, then we say x_o is robust against adversarial perturbations.

We can extend this notion of adversarial robustness to the whole image space $\mathcal{X}$. To do this, we define the *robust region* of a classifier c as the set of images that c can output robust predictions in the presence of adversarial perturbations.

Definition 1 (ϵ-robust region). *Let $c : \mathcal{X} \to \mathcal{Y}$ be a classifier and $\epsilon > 0$ be the perturbation budget of an adversary. The ϵ-robust region of the classifier c is defined by $R_\epsilon(c) := \{x \in \mathcal{X} \mid c(x') = c(x),\ \forall x' \in B_\epsilon(x)\}$.*

Now, consider a defender who can preemptively manipulate x_o under a small modification budget $\delta > 0$ to generate a new image $x_r \in B_\delta(x_o)$, and a grey-box adversary who aims to find an adversarial example near x_r. Then, the defender's optimal strategy against the adversary is to make x_r be correctly classified as y_o and locate in the robust region $R_\epsilon(c)$ so that x_r is robust to adversarial perturbations. If both of these two conditions are satisfied, we say x_o is *preemptively robust* against the grey-box adversary and x_r is a *preemptively robustified image* of x_o.

Definition 2 (Preemptive robustness, grey-box). *Let $c : \mathcal{X} \to \mathcal{Y}$ be a classifier and $\delta, \epsilon > 0$ be the modification budgets of the defender and the grey-box adversary, respectively. An original image x_o with its class y_o is preemptively robust against the grey-box adversary if there exists $x_r \in B_\delta(x_o)$ such that (i) $c(x_r) = y_o$ and (ii) $x_r \in R_\epsilon(c)$.*

Next, we consider a white-box adversary against the defender. Let us denote the manipulation algorithm of the defender by $m : \mathcal{X} \times \mathcal{Y} \to \mathcal{X}$, *i.e.*, $x_r = m(x_o, y_o)$. Then, the white-box adversary will adaptively design its attack algorithm $a_m : \mathcal{X} \times \mathcal{Y} \to \mathcal{X}$, which takes x_r and y_o as inputs and produces a candidate of the adversarial example. For the output $a_m(x_r, y_o)$ to be a valid adversarial example, it should be misclassified and located in $B_\epsilon(x_o)$. If the output is not a valid adversarial example, we say x_o is *preemptively robust* against the white-box adversary.

Definition 3 (Preemptive robustness, white-box). *Let $m : \mathcal{X} \times \mathcal{Y} \to \mathcal{X}$ be the defender's manipulation algorithm and $a_m : \mathcal{X} \times \mathcal{Y} \to \mathcal{X}$ be the adaptive attack algorithm of the white-box adversary. Given an original image x_o and its class y_o, let $x_r = m(x_o, y_o)$ denote the resulting image of the defender's algorithm. Then, x_o is called preemptively robust against the white-box adversary if either of the following conditions is satisfied: (i) $c(a_m(x_r, y_o)) = y_o$ or (ii) $a_m(x_r, y_o) \notin B_\epsilon(x_o)$.*

Note that since the white-box adversary does not have any information about the original x_o in the MitM setting, forcing $a_m(x_r, y_o)$ to lie in $B_\epsilon(x_o)$ is a non-trivial task for the adversary.

Preemptive Robustification Algorithm

In this subsection, we develop an algorithm for preemptively robustifying original images against the grey-box adversary. Given a classifier c, finding a preemptively robustified image x_r from an original image x_o can be formulated as the following optimization problem, which is directly from Definition 2:

$$\begin{aligned} \underset{x_r}{\text{minimize}} \quad & \mathbb{1}_{c(x_r) \neq y_o} + \mathbb{1}_{x_r \notin R_\epsilon(c)} \\ \text{subject to} \quad & \|x_r - x_o\|_p \leq \delta, \end{aligned}$$

where $\mathbb{1}$ is the 0-1 loss function.

Note that in this formulation, the defender requires the ground-truth label y_o. However, images in real-world applications are usually unlabeled unless users manually annotate their images. Therefore, it is natural to assume that the defender does not have access to the ground-truth label y_o. In this case, we utilize the classifier's prediction $c(x_o)$ instead of y_o:

$$\begin{aligned} \underset{x_r}{\text{minimize}} \quad & \mathbb{1}_{c(x_r) \neq c(x_o)} + \mathbb{1}_{x_r \notin R_\epsilon(c)} \\ \text{subject to} \quad & \|x_r - x_o\|_p \leq \delta. \end{aligned}$$

As $x_r \notin R_\epsilon(c)$ implies there exists an adversarial example $x_r^a \in B_\epsilon(x_r)$ such that $c(x_r^a) \neq c(x_r)$, we can reformulate the optimization problem as

$$\begin{aligned} \underset{x_r}{\text{minimize}} \quad & \mathbb{1}_{c(x_r) \neq c(x_o)} + \sup_{x_r^a} \mathbb{1}_{c(x_r^a) \neq c(x_r)} \\ \text{subject to} \quad & \|x_r - x_o\|_p \leq \delta \ \text{ and } \ \|x_r^a - x_r\|_p \leq \epsilon. \end{aligned}$$

Since 0-1 loss is not differentiable, we use the cross-entropy loss $\ell : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}^+$ of the classifier c as the convex surrogate loss function:

$$\begin{aligned} \underset{x_r}{\text{minimize}} \quad & \ell(x_r, c(x_o)) + \sup_{x_r^a} \ell(x_r^a, c(x_r)) \qquad (1) \\ \text{subject to} \quad & \|x_r - x_o\|_p \leq \delta \ \text{ and } \ \|x_r^a - x_r\|_p \leq \epsilon. \end{aligned}$$

Let $h(x_r)$ denote the objective in Equation (1). Instead of minimizing $h(x_r)$ directly, we minimize $\tilde{h}(x_r) = \sup_{x_r^a \in B_\epsilon(x_r)} \ell(x_r^a, c(x_o))$ since it upper bounds $h(x_r)$ when sufficiently minimized due to Lemma 1.

Lemma 1. *If $\tilde{h}(x_r) \leq -\log(0.5) \simeq 0.6931$, then $h(x_r) \leq 2\tilde{h}(x_r)$.*

Proof. See Supplementary A.1. □

Finally, we have the following optimization problem:

$$\begin{aligned} \underset{x_r}{\text{minimize}} \quad & \sup_{x_r^a} \ell(x_r^a, c(x_o)) \qquad (2) \\ \text{subject to} \quad & \|x_r - x_o\|_p \leq \delta \ \text{ and } \ \|x_r^a - x_r\|_p \leq \epsilon. \end{aligned}$$

To solve Equation (2), we first approximate the inner maximization problem by running T-step PGD (Madry et al. 2017) whose dynamics is given by

$$\begin{aligned} x_r^{a,0} &= x_r + \eta & \text{(random start)} \\ \tilde{x}_r^{a,t} &= f\left(x_r^{a,t-1}; c(x_o), \ell\right) & \text{(adversarial update)} \\ x_r^{a,t} &= \Pi_{x_r,\epsilon}\left(\tilde{x}_r^{a,t}\right), & \text{(projection)} \end{aligned}$$

Algorithm 1: Preemptive robustification algorithm

```
input Original image and its prediction $(x_o, c(x_o))$
  $x_r = x_o$ // or randomly initialized in $B_\delta(x_o)$
  for $i = 1, \ldots,$ MAXITER do
    // Generate $N$ PGD adversarial examples
    for $n = 1, \ldots, N$ do
      $x^a_{r,n} = x_r + \eta$ where $\eta \sim \mathcal{U}(B_\epsilon(0))$
      for $t = 1, \ldots, T$ do
        $x^a_{r,n} \leftarrow \Pi_{x_r,\epsilon}\left(f(x^a_{r,n}; c(x_o), \ell)\right)$
      end for
    end for
    // Update image
    $x_r \leftarrow \Pi_{x_o,\delta}\left(x_r - \beta \cdot \frac{1}{N}\sum_{n=1}^{N} \frac{\partial \ell(x^a_{r,n}, c(x_o))}{\partial x_r}\right)$
  end for
output $x_r$
```

where η is a noise uniformly sampled from $B_\epsilon(0)$, f is FGSM (Goodfellow, Shlens, and Szegedy 2015) defined as

$$f(x; y, \ell) = \begin{cases} x + \alpha \cdot \mathrm{sgn}\left(\nabla_x \ell(x, y)\right) & \text{if } p = \infty \\ x + \alpha \cdot \dfrac{\nabla_x \ell(x, y)}{\|\nabla_x \ell(x, y)\|_2} & \text{if } p = 2, \end{cases}$$

and $\Pi_{x_r,\epsilon}$ is a projection operation onto $B_\epsilon(x_r)$. Then, we iteratively solve the approximate problem given by replacing x^a_r to $x^{a,T}_r$ in Equation (2). To update x_r, we compute the gradient of $\ell(x^a_r, c(x_o))$ with respect to x_r expressed as

$$\frac{\partial \ell\left(x^a_r, c(x_o)\right)}{\partial x_r} = \nabla_x \tilde{f}\left(x^{a,0}_r\right)^\intercal \cdot \cdots \cdot \nabla_x \tilde{f}\left(x^{a,T-1}_r\right)^\intercal \cdot \nabla_x \ell\left(x^{a,T}_r, c(x_o)\right),$$

where $\nabla_x \tilde{f}$ is the Jacobian matrix of $\tilde{f} = \Pi_{x_r,\epsilon} \circ f$ which can be computed via back-propagation. After computing the gradient, we update x_r by projected gradient descent method:

$$x_r \leftarrow \Pi_{x_o,\delta}\left(x_r - \beta \cdot \frac{\partial \ell(x^a_r, c(x_o))}{\partial x_r}\right).$$

Note that $\ell(x^a_r, c(x_o))$ is a random variable dependent on η. Therefore, we generate N adversarial examples $\{x^a_{r,n}\}^N_{n=1}$ with different noises and optimize the sample mean of the losses instead. Algorithm 1 shows the overall preemptive robustification algorithm and Figure 2 illustrates the optimization process. Some examples of robustified images generated from our algorithm are shown in Supplementary D.

Computing Update Gradient without Second-Order Derivatives

Computing the update gradient with respect to x_r involves the use of second-order derivatives of the loss function ℓ since the dynamics f contains the loss gradient $\nabla_x \ell$. Standard deep learning libraries, such as PyTorch (Paszke et al. 2019), support the computation of these higher-order derivatives. However, it imposes a huge memory burden as the size of the computational graph increases. Furthermore, when $p = 2$, computing the update gradient with the second-order derivatives might cause an *exploding gradient problem* if the loss gradient vanishes by Lemma 2 and Proposition 1.

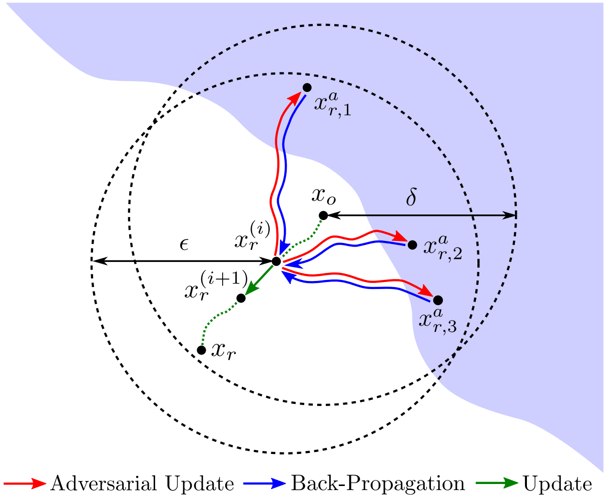

Figure 2: Illustration of the preemptive robustification process. The shaded region represents the set of misclassified points.

Lemma 2. *Suppose ℓ is twice-differentiable and its second partial derivatives are continuous. If $p = 2$, the Jacobian of the dynamics f is*

$$\nabla_x f = I + \alpha \cdot \left(I - \left(\frac{g}{\|g\|_2}\right)\left(\frac{g}{\|g\|_2}\right)^\intercal\right)\frac{H}{\|g\|_2},$$

where $g = \nabla_x \ell$ and $H = \nabla^2_x \ell$.

Proof. See Supplementary B.1. □

Proposition 1. *If the maximum eigenvalue of H in absolute value is σ, then*

$$\left\|\nabla_x \tilde{f}^\intercal \cdot a\right\|_2 \leq \left(1 + \alpha \cdot \frac{\sigma}{\|g\|_2}\right)\|a\|_2.$$

Proof. See Supplementary B.2. □

As we update x_r, the loss gradients g of x_r and its intermediate adversarial examples $x^{a,t}_r$ reduce to zero, which might cause the update gradient to explode and destabilize the update process. To address this problem, we approximate the update gradient by excluding the second-order derivatives, following the practice in Finn, Abbeel, and Levine (2017). We also include an experiment in comparison to using the exact update gradient in Supplementary B.3. For the case of $p = \infty$, the second-order derivatives naturally vanish since we take the sign of the loss gradient $\nabla_x \ell(x, y)$. Therefore, the approximate gradient is equal to the exact update gradient.

Network Training Scheme for Improving Preemptive Robustness

So far, we have explored how the defender can preemptively robustify the original image, given a pre-trained classifier. Now, we explore the defender's network training scheme for a classifier where data points are preemptively robust with high probability. Suppose the defender has a labeled training set, which is drawn from a true data distribution $\mathcal{D}$. To induce data points to be preemptively robust, the defender's optimal training objective should have the following form:

$$\begin{aligned} \underset{\theta}{\text{minimize}} \quad & \underset{(x_o,y_o)\sim\mathcal{D}}{\mathbb{E}} \left[\ell(\hat{x}_r^a, y_o;\theta)\right] \\ \text{subject to} \quad & \hat{x}_r^a = \underset{x_r^a\in B_\epsilon(\hat{x}_r)}{\text{argmax}}\ \ell(x_r^a,y_o) \\ & \hat{x}_r = \underset{x_r\in B_\delta(x_o)}{\text{argmin}}\ \sup_{x_r^a\in B_\epsilon(x_r)} \ell(x_r^a,y_o), \end{aligned}$$

where θ is the set of trainable parameters. Concretely, the defender first attempts to craft a candidate for preemptively robustified points $\hat{x}_r$ of the original data point x_o. Then, the defender generates an adversarial example $\hat{x}_r^a$ of $\hat{x}_r$ and minimizes its cross-entropy loss $\ell(\hat{x}_r^a, y_o;\theta)$ so that $\hat{x}_r$ becomes resistant to adversarial perturbations. Note that the ground-truth label y_o is used instead of the prediction $c(x_o)$, since we assume the ground-truth label of the training set is given.

The most direct way to optimize the objective would be to find $\hat{x}_r$ from x_o using our preemptive robustification algorithm and perform K-step PGD adversarial training (Madry et al. 2017) with $\hat{x}_r$. However, since our algorithm requires running T-step PGD dynamics per each update, the proposed training procedure would be more computationally demanding than standard PGD adversarial training. To ease this problem, we replace the inner maximization $\sup_{x_r^a} \ell(x_r^a, y_o)$ in the preemptive robustification process by $\ell(x_r, y_o)$:

$$\begin{aligned} & \hat{x}_r = \underset{x_r\in B_\delta(x_o)}{\text{argmin}}\ \sup_{x_r^a\in B_\epsilon(x_r)} \ell(x_r^a,y_o) \\ \Longrightarrow\ & \hat{x}_r = \underset{x_r\in B_\delta(x_o)}{\text{argmin}}\ \ell(x_r,y_o). \end{aligned}$$

Then, $\hat{x}_r$ can be easily computed by running L-step PGD on x_o towards minimizing the cross-entropy loss. We denote this training scheme as *preemptively robust training*. The full training procedure is summarized in Algorithm 2 (differences with the standard adversarial training marked in blue).

Algorithm 2: Preemptively robust training, $p = \infty$

```
input Training dataset $\mathcal{D}_{train}$, maximum epoch $N$
  for $n = 1, \dots, N$ do
    for $(x_o, y_o) \in \mathcal{D}_{train}$ do
      // Do $L$-step PGD towards minimizing loss
      $x_r = x_o + \eta$ where $\eta \sim B_\delta(0)$
      for $l = 1, \dots, L$ do
        $x_r \leftarrow \Pi_{x_o,\delta}\left(x_r - \beta \cdot \mathrm{sgn}(\nabla_x \ell(x_r, y_o))\right)$
      end for
      // Do $K$-step PGD towards maximizing loss
      $x_r^a = x_r + \eta$ where $\eta \sim B_\epsilon(0)$
      for $t = 1, \dots, K$ do
        $x_r^a \leftarrow \Pi_{x_r,\epsilon}\left(x_r^a + \alpha \cdot \mathrm{sgn}(\nabla_x \ell(x_r^a, y_o))\right)$
      end for
      $\theta \leftarrow \theta - \nabla_\theta \ell(x_r^a, y_o)$
    end for
  end for
```

Note that the standard adversarial training is a specific case of our preemptively robust training, forcing training data x_o to be far from the decision boundary. However, recent work demonstrates there is a trade-off between the classification error and the boundary error, which is why standard adversarial training significantly decreases the clean accuracy (Tsipras et al. 2019; Zhang et al. 2019b). In contrast, our proposed training scheme allows original images x_o to lie near the decision boundary and only enforces the preemptively robustified images $\hat{x}_r$ to be distant from the boundary. Our experiments in **??** show preemptively robust training is less prone to suffer from the clean accuracy drop due to this flexibility.

Preemptive Robustification for Classifiers with Randomized Smoothing

Our preemptive robustification algorithm can also be applied to smoothed classifiers. Given a base classifier $c : \mathcal{X} \to \mathcal{Y}$, a smoothed classifier $\tilde{c} : \mathcal{X} \to \mathcal{Y}$ is defined as

$$\tilde{c}(x) = \underset{y\in\mathcal{Y}}{\text{argmax}}\ \mathbb{P}\left(c(x+\xi) = y\right),$$

where $\xi \sim \mathcal{N}(0, \sigma^2 I)$. Crafting adversarial examples x_r^a for the smoothed classifier, which is necessary for approximating the inner maximization in Equation (2), is ill-behaved since the argmax operation is non-differentiable. To address this problem, we follow the practice in Salman et al. (2019) and approximate the smoothed classifier $\tilde{c}$ with the smoothed soft classifier $\tilde{C} : \mathcal{X} \to P(\mathcal{Y})$ defined as

$$\tilde{C}(x) = \underset{\xi\sim\mathcal{N}(0,\sigma^2 I)}{\mathbb{E}} \left[C(x+\xi)\right], \tag{3}$$

where $P(\mathcal{Y})$ is the set of probability distribution over $\mathcal{Y}$ and $C : \mathcal{X} \to P(\mathcal{Y})$ is the soft version of the base classifier c such that $\text{argmax}_{y\in\mathcal{Y}}\, C(x)_y = c(x)$. Finally, the adversarial example x_r^a can be found by maximizing the cross-entropy loss of $\tilde{C}$ instead:

$$\begin{aligned} \underset{x_r^a}{\text{maximize}} \quad & -\log\left(\tilde{C}(x_r^a)_{c(x_o)}\right) \\ \text{subject to} \quad & \|x_r^a - x_r\|_p \le \epsilon, \end{aligned}$$

which can be approximated by T-step randomized PGD (Salman et al. 2019), where ξ is sampled M times to compute the sample mean of Equation (3) at each step. By replacing the inner maximization problem in Equation (2) with the randomized PGD, we can update x_r in a similar process.

Adaptive Attack against Preemptive Robustification

So far, we have developed preemptive defense strategies against the grey-box adversary. Now, we consider the white-box adversary described in **??** , which is aware that the given image x_r has been preemptively modified and aims to craft an adversarial example near the original image x_o that is unknown. The most direct way for the adversary to achieve this is to reconstruct x_o from x_r and apply standard attack algorithms (*e.g.*, PGD) on the reconstructed image $\hat{x}_o$. Since we assume the adversary knows the detailed hyperparameter settings of the robustification algorithm, the adversary can leverage this information to approximate the inverse dynamics of the preemptive robustification process starting from x_r, as described in Algorithm 3. The only difference between this

Algorithm 3: Original image reconstruction

```
input Preemptively robustified image and its prediction
  $(x_r, c(x_r))$
  $\hat{x}_o = x_r$
  for $i = 1, \ldots, \mathrm{MAXITER}$ do
    // Generate $N$ adversarial examples
    for $n = 1, \ldots, N$ do
      $\hat{x}^a_{o,n} = \hat{x}_o + \eta$ where $\eta \sim \mathcal{U}(B_\epsilon(0))$
      for $t = 1, \ldots, T$ do
        $\hat{x}^a_{o,n} \leftarrow \Pi_{\hat{x}_o,\epsilon}\left(f(\hat{x}^a_{o,n}; c(x_r), \ell)\right)$
      end for
    end for
    // Update image
    $\hat{x}_o \leftarrow \Pi_{x_r,\delta}\left(\hat{x}_o + \beta \cdot \frac{1}{N}\sum_{n=1}^{N} \frac{\partial \ell(\hat{x}^a_{o,n}, c(x_r))}{\partial \hat{x}_o}\right)$
  end for
output $\hat{x}_o$
```

Algorithm 4: Adaptive white-box attack

```
input Preemptively robustified image and its prediction
  $(x_r, c(x_r))$, target $y_o$
  // Reconstruct original image
  $\hat{x}_o = \textsc{OriginalImageReconstruction}(x_r, c(x_r))$
  // Run standard attack algorithm on reconstructed image
  $\hat{x}^a_o = \textsc{AttackAlgorithm}(\hat{x}_o, y_o, \epsilon')$
output $\hat{x}^a_o$
```

reconstruction algorithm and the preemptive robustification process is the initialization and the update direction, modified to suit the reconstruction objective. Algorithm 4 shows the overall procedure of the adaptive white-box attack. Note that the adversary may modify $\hat{x}_o$ with a budget smaller than ϵ to induce $\hat{x}^a_o \in B_\epsilon(x_o)$, considering the original image might not be reconstructed accurately.

Figure 3 shows the proposed reconstruction algorithm performs well in terms of reconstruction error if the preemptive robustification algorithm starts from the original image itself. However, as we run the preemptive robustification algorithm starting from a random point in $B_\delta(x_o)$, the performance of the reconstruction algorithm degrades considerably. About 80% of the reconstructed images locate near the boundaries of ϵ-balls centered at original images, which shows the difficulty of reconstructing the original image. We also observe that most of the resulting white-box attack examples generated from the reconstructed images lie outside $B_\epsilon(x_o)$, which implies they are not valid adversarial examples.

Experiments

We evaluate our methods on CIFAR-10 and ImageNet by measuring classification accuracies of preemptively robustified images under the grey-box and white-box adversaries. As it is natural to assume that the defender and the adversary have the same modification budget, we set $\delta = \epsilon$ for all experiments. Both the adversaries use 20-step untargeted PGD and AutoAttack (Croce and Hein 2020) to find adversarial examples. For the white-box adversary, we sweep the final perturbation budget ϵ' and report the lowest accuracy measured. More details are listed in Supplementary C. The code is available online [1].

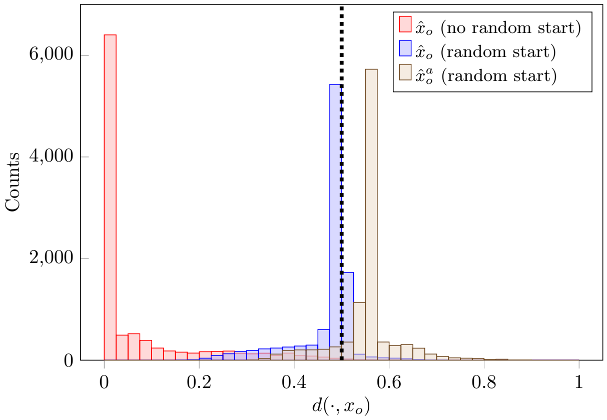

Figure 3: Histograms of the distances between original images x_o and the reconstructed images $\hat{x}_o$ or their white-box attack examples $\hat{x}^a_o$ on CIFAR-10 test set. We use a preemptively robust model with $p = 2$ and $\delta = \epsilon = 0.5$. The dotted line indicates the adversary's perturbation bound ϵ.

CIFAR-10

We consider two types of perturbations: ℓ_∞ with $\epsilon = 8/255$ and ℓ_2 with $\epsilon = 0.5$. To show the effectiveness of our robustification algorithm, we also report results without preemptive robustification (None)[2]. As the baseline for our preemptively robust model, we use an adversarially trained model with early stopping (Rice, Wong, and Kolter 2020) (ADV).

The results in Table 1 and Table 2 show that our preemptive manipulation method successfully robustifies images compared to without the manipulation, achieving adversarial accuracies higher than 80% in both the perturbation settings. Also, while the adversarially trained model does induce some images to be preemptively robust, our proposed network training method further boosts the performance of the robustified images in both the clean and adversarial accuracies.

Note that the white-box attacks are not effective on ℓ_∞ as they fail to lay their adversarial examples within the $B_\epsilon(x_o)$ ball. This is in part due to the characteristic of the ℓ_∞ distance measure, as ℓ_∞ distance can spike even when a single pixel deviates from the original value. On the other hand, white-box attacks on ℓ_2 perturbations succeed to pose reasonable threats. However, the worst-case adversarial accuracy is still over 10% higher than without our methods.

ImageNet

We consider two types of perturbations: ℓ_∞ with $\epsilon = 4/255$ and ℓ_2 with $\epsilon = 3.0$. As with the CIFAR-10 experiments, we compare our preemptive robustification algorithm to without

[1] https://github.com/snu-mllab/preemptive_robustification

[2] In this case, the grey-box adversary is the same as the white-box adversary since no modification occurs on original images.

Model	Preempt.	Clean	Grey-box		White-box	
			PGD	AA	PGD	AA
ADV	None	86.72	54.59	51.68	54.59	51.68
ADV	**Ours**	86.72	86.23	81.70	86.72	86.72
Ours	**Ours**	**88.54**	**87.10**	**82.88**	**88.54**	**88.54**

Table 1: CIFAR-10 classification accuracy under grey-box and white-box adversaries with ℓ_∞ perturbation, $\epsilon = 8/255$.

Model	Preempt.	Clean	Grey-box		White-box	
			PGD	AA	PGD	AA
ADV	None	90.85	71.90	71.21	71.90	71.21
ADV	**Ours**	90.85	84.81	83.56	**85.12**	79.48
Ours	**Ours**	**92.57**	**91.81**	**89.32**	85.02	**80.79**

Table 2: CIFAR-10 classification accuracy under grey-box and white-box adversaries with ℓ_2 perturbation, $\epsilon = 0.5$.

the algorithm (None). As the baseline, we use a model adversarially trained with the fast training schemes (Wong, Rice, and Kolter 2020) for computational efficiency (ADV).

Table 3 and Table 4 show our preemptive robustification methods scale to more practical datasets with bigger images. Our methods allow the natural images to be much more robust to adversarial attacks, with over 15% higher worst-case adversarial accuracies, and at the same time maintains higher clean accuracies. We observe that similar to the CIFAR-10 experiments, the white-box attacks on the ℓ_∞ distance measure are not successful in finding appropriate adversarial samples.

Model	Preempt.	Clean	Grey-box		White-box	
			PGD	AA	PGD	AA
ADV	None	56.24	32.03	27.52	32.03	27.52
ADV	**Ours**	56.24	55.79	47.14	56.24	56.24
Ours	**Ours**	**61.01**	**59.66**	**48.24**	**61.01**	**61.01**

Table 3: ImageNet classification accuracy under grey-box and white-box adversaries with ℓ_∞ perturbation, $\epsilon = 4/255$.

Model	Preempt.	Clean	Grey-box		White-box	
			PGD	AA	PGD	AA
ADV	None	54.99	32.07	27.58	32.07	27.58
ADV	**Ours**	55.05	51.70	43.32	46.38	37.49
Ours	**Ours**	**61.60**	**58.13**	**43.60**	**54.23**	**47.54**

Table 4: ImageNet classification accuracy under grey-box and white-box adversaries with ℓ_2 perturbation, $\epsilon = 3.0$.

Randomized Smoothing

We also evaluate our preemptive robustification algorithm for smoothed classifiers. We consider ℓ_2 perturbations, where $\epsilon = 0.5$ for CIFAR-10 and $\epsilon = 3.0$ for ImageNet. We utilize a smoothed model trained with Gaussian noise augmentation as proposed in Cohen, Rosenfeld, and Kolter (2019) due to its simplicity. We measure empirical adversarial accuracies using 20-step randomized PGD and its 10 restart version (PGD-10). We also compute the certified radii of the images and measure the certified adversarial accuracies.

Table 5 and Table 6 shows the empirical robustness results against the randomized PGD. We observe our methods can significantly enhance preemptive robustness on the smoothed classifiers, maintaining 28% and 49% higher the worst-case adversarial accuracies than the baseline on CIFAR10 and ImageNet, respectively. The results in Table 7 show our method also improves the certified robustness on the smoothed networks. Our methods achieve 22% and 15% higher certified accuracies on CIFAR10 and ImageNet, respectively.

Preempt.	Clean	Grey-box		White-box	
		PGD	PGD-10	PGD	PGD-10
None	92.14	56.02	53.01	56.02	53.01
Ours	**92.35**	**91.37**	**89.98**	**82.06**	**80.71**

Table 5: CIFAR-10 empirical accuracy of smoothed network under grey-box and white-box adversaries with ℓ_2 perturbation, $\epsilon = 0.5$. We set the noise level to $\sigma = 0.1$.

Preempt.	Clean	Grey-box		White-box	
		PGD	PGD-10	PGD	PGD-10
None	69.93	9.61	8.61	9.61	8.61
Ours	**70.05**	**62.27**	**57.24**	**68.05**	**67.72**

Table 6: ImageNet empirical accuracy of smoothed network under grey-box and white-box adversaries with ℓ_2 perturbation, $\epsilon = 3.0$. We set the noise level to $\sigma = 0.25$.

Preempt.	Clean	Cert.
None	82.84	55.58
Ours	**84.72**	**77.95**

Preempt.	Clean	Cert.
None	47.02	12.68
Ours	**52.66**	**27.89**

Table 7: CIFAR-10 (left) and ImageNet (right) certified accuracies of smoothed network with ℓ_2 perturbation. The noise levels are $\sigma = 0.25$ for CIFAR-10 and $\sigma = 1.0$ for ImageNet.

Conclusion

We consider a real-world adversarial framework where the MitM adversary intercepts and manipulates the images during transmission. To protect users from such attacks, we introduce a novel optimization algorithm for finding robust points in the vicinity of original images along with a new network training method suited for enhancing preemptive robustness. The experiments show that our algorithm can find such robust points for most of the correctly classified images. Further results show our method also improves preemptive robustness on smooth classifiers.

Acknowledgements

This work was supported in part by SNU-NAVER Hyperscale AI Center and Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. 2020-0-00882, (SW STAR LAB) Development of deployable learning intelligence via self-sustainable and trustworthy machine learning and No. 2019-0-01371, Development of brain-inspired AI with human-like intelligence). This material is based upon work supported by the Air Force Office of Scientific Research under award number FA2386-20-1-4043.

References

Cherepanova, V.; Goldblum, M.; Foley, H.; Duan, S.; Dickerson, J.; Taylor, G.; and Goldstein, T. 2021. LowKey: leveraging adversarial attacks to protect social media users from facial recognition. In *ICLR*.

Cohen, J. M.; Rosenfeld, E.; and Kolter, J. Z. 2019. Certified adversarial robustness via randomized smoothing. In *ICML*.

Croce, F.; and Hein, M. 2020. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In *ICML*.

Desmedt, Y. 2011. *Man-in-the-Middle Attack*. Springer.

Finn, C.; Abbeel, P.; and Levine, S. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. In *ICML*.

Goodfellow, I. J.; Shlens, J.; and Szegedy, C. 2015. Explaining and harnessing adversarial examples. In *ICLR*.

Krizhevsky, A.; and Hinton, G. 2009. Learning multiple layers of features from tiny images. Technical report, Citeseer.

Lecuyer, M.; Atlidakis, V.; Geambasu, R.; Hsu, D.; and Jana, S. 2019. Certified robustness to adversarial examples with differential privacy. In *IEEE Symposium on Security and Privacy (SP)*.

Li, S.; Neupane, A.; Paul, S.; Song, C.; Krishnamurthy, S. V.; Roy-Chowdhury, A. K.; and Swami, A. 2019. Adversarial Perturbations Against Real-Time Video Classification Systems. In *NDSS*.

Madry, A.; Makelov, A.; Schmidt, L.; Tsipras, D.; and Vladu, A. 2017. Towards deep learning models resistant to adversarial attacks. In *ICLR*.

Oh, S. J.; Fritz, M.; and Schiele, B. 2017. Adversarial image perturbation for privacy protection a game theory perspective. In *ICCV*.

Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; et al. 2019. PyTorch: An imperative style, high-performance deep learning library. In *NeurIPS*.

Raghunathan, A.; Steinhardt, J.; and Liang, P. 2018. Certified defenses against adversarial examples. In *ICLR*.

Rice, L.; Wong, E.; and Kolter, J. Z. 2020. Overfitting in adversarially robust deep learning. In *ICML*.

Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; Berg, A. C.; and Fei-Fei, L. 2015. ImageNet Large Scale Visual Recognition Challenge. In *IJCV*.

Salman, H.; Ilyas, A.; Engstrom, L.; Vemprala, S.; Madry, A.; and Kapoor, A. 2020. Unadversarial Examples: Designing Objects for Robust Vision. arXiv:2012.12235.

Salman, H.; Li, J.; Razenshteyn, I.; Zhang, P.; Zhang, H.; Bubeck, S.; and Yang, G. 2019. Provably robust deep learning via adversarially trained smoothed classifiers. In *NeurIPS*.

Shafahi, A.; Najibi, M.; Ghiasi, A.; Xu, Z.; Dickerson, J.; Studer, C.; Davis, L.; Taylor, G.; and Goldstein, T. 2019. Adversarial Training for Free! In *NeurIPS*.

Shan, S.; Wenger, E.; Zhang, J.; Li, H.; Zheng, H.; and Zhao, B. Y. 2020. Fawkes: Protecting privacy against unauthorized deep learning models. In *29th USENIX Security Symposium (USENIX Security 20)*.

Szegedy, C.; Zaremba, W.; Sutskever, I.; Bruna, J.; Erhan, D.; Goodfellow, I.; and Fergus, R. 2013. Intriguing properties of neural networks. In *ICLR*.

Tsipras, D.; Santurkar, S.; Engstrom, L.; Turner, A.; and Madry, A. 2019. Robustness may be at odds with accuracy. In *ICLR*.

Wang, D.; Li, C.; Wen, S.; Nepal, S.; and Xiang, Y. 2019. Man-in-the-Middle Attacks against Machine Learning Classifiers via Malicious Generative Models. In *IEEE Transactions on Dependable and Secure Computing*.

Wong, E.; and Kolter, Z. 2018. Provable defenses against adversarial examples via the convex outer adversarial polytope. In *ICML*.

Wong, E.; Rice, L.; and Kolter, J. Z. 2020. Fast is better than free: Revisiting adversarial training. In *ICLR*.

Yang, G.; Duan, T.; Hu, J. E.; Salman, H.; Razenshteyn, I.; and Li, J. 2020. Randomized smoothing of all shapes and sizes. In *ICML*.

Zhang, D.; Zhang, T.; Lu, Y.; Zhu, Z.; and Dong, B. 2019a. You only propagate once: Accelerating adversarial training via maximal principle. In *NeurIPS*.

Zhang, H.; and Wang, J. 2019. Defense against adversarial attacks using feature scattering-based adversarial training. In *NeurIPS*.

Zhang, H.; Yu, Y.; Jiao, J.; Xing, E. P.; Ghaoui, L. E.; and Jordan, M. I. 2019b. Theoretically Principled Trade-off between Robustness and Accuracy. In *ICML*.

Zhang, J.; Xu, X.; Han, B.; Niu, G.; Cui, L.; Sugiyama, M.; and Kankanhalli, M. 2020. Attacks Which Do Not Kill Training Make Adversarial Learning Stronger. In *ICML*.