PatchGuard: Provable Defense against Adversarial Patches Using Masks on Small Receptive Fields

Chong Xiang, Arjun Nitin Bhagoji, Vikash Sehwag, Prateek Mittal
Princeton University
{cxiang, abhagoji, vvikash, pmittal}@princeton.edu

Abstract—Localized adversarial patches aim to induce misclassification in machine learning models by arbitrarily modifying pixels within a restricted region of an image. Such attacks can be realized in the physical world by attaching the adversarial patch to the object to be misclassified. In this paper, we propose a general defense framework that can achieve both high clean accuracy and provable robustness against localized adversarial patches. The cornerstone of our defense framework is to use a convolutional network with small receptive fields that impose a bound on the number of features corrupted by an adversarial patch. We further present the robust masking defense that robustly detects and masks corrupted features for a secure feature aggregation. We evaluate our defense against the most powerful white-box untargeted adaptive attacker and achieve a 92.3% clean accuracy and an 85.2% provable robust accuracy on a 10-class subset of ImageNet against a 31x31 adversarial patch (2% pixels), a 57.4% clean accuracy and a 14.4% provable robust accuracy on 1000-class ImageNet against a 31x31 patch (2% pixels), and an 80.3% clean accuracy and a 61.3% provable accuracy on CIFAR-10 against a 5x5 patch (2.4% pixels). Notably, our provable defenses achieve state-of-the-art provable robust accuracy on ImageNet and CIFAR-10.

I. INTRODUCTION

Machine learning models are vulnerable to evasion attacks, where an adversary introduces a small perturbation to a test example for inducing model misclassification [1], [2]. Many prior attacks and defenses focus on the classical setting of adversarial examples that have a small $L_p$ distance to a normal example [1]–[12]. However, in the physical world, this classical $L_p$ setting may require global perturbations to an object, which is not always practical. In this paper, we focus on the threat of localized adversarial patches, in which the adversary can arbitrarily modify pixels within a restricted area such that the perturbation can be realized by attaching an adversarial patch to the victim object. Brown et al. [13] generate physical adversarial patches that can universally hijack the model prediction to a targeted class. Karmon et al. [14] propose LaVAN attacks that craft non-universal targeted/untargeted adversarial patches within the digital domain. Eykholt et al. [15] demonstrate a robust physical world attack that attaches small stickers to a stop sign for fooling traffic sign classification.

The success of practical localized adversarial patches has inspired several defense approaches. Digital Watermark (DW) [16] aims to detect and remove the adversarial patch. Local Gradient Smoothing (LGS) [17] proposes to smooth the suspicious region of pixels to neutralize the adversarial patch. However, these empirical defenses are heuristic approaches and lack robustness against a strong adaptive attacker [18]. Chiang et al. [18] propose the first certified defense against adversarial patch via Interval Bound Propagation (IBP) [19], [20]. Zhang et al. [21] use clipped BagNet to achieve provable robustness, and Levine et al. [22] propose De-randomized Smoothing (DS) to further improve the provable robustness. While these works have made important contributions, prior certified defenses usually struggle with poor clean accuracy and provable robustness.

In this paper, we propose a general defense framework called PatchGuard that achieves high clean accuracy and provable robustness against localized adversarial patches. The cornerstone of our defense framework is to use a convolutional network with small receptive fields that impose a bound on the number of corrupted features due to an adversarial patch. The receptive field of a convolutional network is the region of an input image that a particular feature is looking at, and the model prediction is based on the aggregation of features extracted from different regions of an image. One example of the receptive field is shown as the red box in the image in Figure 1. Our case study in Section III-A shows that a large receptive field makes convolutional networks more vulnerable to adversarial patch attacks. For a model with a large receptive field of 483x483 (ResNet-50 [23]) on ImageNet images [24], a small patch with 2% corrupted image pixels will appear in the receptive field of all extracted features and can thus easily change the model prediction. A small receptive field, on the other hand, can limit the number of corrupted features, and we use it as a fundamental building block of robustness. We note that a small receptive field is not a barrier to achieving high clean accuracy. A ResNet-like architecture with a 17x17 receptive field can achieve an AlexNet-level accuracy for ImageNet top-5 classification [25]. The potential robustness improvement, as well as the moderate accuracy drop, motivates the use of small receptive fields in our defense framework.

However, a small receptive field alone is not enough for a robust prediction due to the insecure feature aggregation (e.g., average) in conventional models. In this paper, we further present robust masking for secure feature aggregation. Figure 1 provides an overview of our defense. The small receptive field provides a robust foundation such that only a small fraction of extracted features are corrupted due to an adversarial patch. A small number of corrupted features require the adversary to create abnormally large feature values to dominate the
Fig. 1. Defense overview. The small receptive field bounds the fraction of corrupted features (one out of three vectors in this example). The only one corrupted feature (red vector) in this example has an abnormally large element which dominates the insecure aggregation ($\Sigma$) but also leads to a distinct pattern from clean features. Our robust masking aims to detect and mask the corrupted feature and recover the correct prediction from the remaining features.

---

II. PROBLEM FORMULATION

In this section, we list the notation and terminology used in this paper and formulate the localized adversarial patch attack, including the attacker’s objective, capability, knowledge, and the algorithm used for adversarial patch generation.

A. Notation and Terminology

We focus on fully convolutional neural networks (CNNs) such as ResNet [23], which use convolutional layers only for feature extraction and an additional fully-connected layer for the final prediction.

Table I provides a summary of our notation. We use $\mathcal{X} \subset [0, 1]^{W \times H \times C}$ to denote the image space where each image has width $W$, height $H$, channel $C$, and the pixel values are re-scaled to $[0, 1]$. We take $\mathcal{Y} = \{0, 1, \cdots, N-1\}$ as the class label space, where the number of classes is $N$. We use $\mathcal{M}(x): \mathcal{X} \rightarrow \mathcal{Y}$ to denote the deep learning model that takes an image $x \in \mathcal{X}$ as input and predicts the class label $y \in \mathcal{Y}$. Let $\mathcal{F}(x): \mathcal{X} \rightarrow \mathcal{U}$ be the feature extractor that outputs feature tensor $u \in \mathcal{U} \subset \mathbb{R}^{W' \times H' \times C'}$, where $W'$, $H'$, $C'$ are the width, height, and channel dimension of this feature map, respectively. We refer to each $C'$-dimensional feature in tensor $u$ as a local feature $\hat{u}$ since it is only extracted from part of the input image as opposed to the entire image. The concept of a feature is general here and can refer to a hidden layer representation, a

---

Table I

| Notation | Description |
|----------|-------------|
| $\mathcal{X}$ | Image space |
| $\mathcal{Y}$ | Label space |
| $\mathcal{U}$ | Feature space |
| $\mathcal{M}(x)$ | Model predictor from $x \in \mathcal{X}$ |
| $\mathcal{M}(u)$ | Model predictor from $u \in \mathcal{U}$ |
| $\mathcal{F}(x)$ | Local feature extractor for all classes |
| $\mathcal{F}(x, l)$ | Local feature extractor for class $l$ |
| $P \subset \{0, 1\}^{W \times H}$ | Set of binary image blocks in image space |
| $W \subset \{0, 1\}^{W' \times H'}$ | Set of binary windows in feature space |

---

final prediction, and our robust masking aims to detect and mask the abnormal features. Our empirical analysis shows that removing a small number of features of a clean image rarely changes model prediction. Therefore, we are likely to recover the correct prediction if all the corrupted features are removed. Robust masking puts the adversary in a dilemma where it will either be detected by our defense or fail to generate an effective adversarial patch. This dilemma further leads to a provable robustness of our defense. Our provable defense provides the guarantee that the model can always make correct predictions on certified images for any adversarial patches. We note that this is a stronger notation of robustness compared with defenses that only detect the adversarial attack [7], [8], [26].

We consider the strongest adversarial patch attack threat model, where the adversarial patch can be placed on any part of the image, including on top of salient objects, and evaluate our defense against the powerful white-box untargeted adaptive attacker on a 10-class subset of ImageNet [27], 1000-class ImageNet [24] and CIFAR-10 [28]. We achieve a 92.3% clean accuracy and an 85.2% provable robust accuracy on a 10-class subset of ImageNet against a 31x31 adversarial patch (2% pixels), a 57.4% clean accuracy and 14.4% provable robust accuracy on 1000-class ImageNet against a 31x31 patch (2% pixels), and an 80.3% clean accuracy and a 61.7% provable accuracy on CIFAR-10 against a 5x5 patch (2.4% pixels). Our defense achieves state-of-the-art provable robustness on ImageNet and CIFAR-10 compared with previous defenses [18], [21], [22]. Our main contributions can be summarized as follows.

1) We demonstrate the use of a small receptive field as a fundamental robustness building block and integrate it in a general defense framework called PatchGuard that mitigates localized adversarial attacks.

2) We present the robust masking defense which has provable robustness and recovers correct predictions for certified images against any white-box adaptive attacker.

3) We provide a comprehensive analysis of our defenses on ImageNet and CIFAR-10 images and demonstrate state-of-the-art provable robust accuracy.
logits tensor, a confidence tensor, or a prediction tensor. When the feature is one of logits (i.e., the output before softmax operation), confidence, or prediction, we have \( C' = N \). In this case, we will sometimes abuse this notation by letting \( F(x, l) : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}^{W' \times H'} \) denote the slice of the feature corresponding to class \( l \). We call the elements of \( F(x, l) \) the class evidence for class \( l \). We will also reuse the notation \( \mathcal{M} \) as a mapping from a hidden feature map space \( \mathcal{U} \) to the label space \( \mathcal{Y} \). \( \mathcal{M}(u) \) in this context implies doing model inference with a feature tensor \( u \). We define the receptive field of a particular feature in a convolutional network to be a subset of image pixels that a particular local feature \( \hat{u} \) is looking at, or affected by. Formally, if we represent the input images \( x \) as a set of pixels, the receptive field of a particular local feature \( \hat{u} \) is a subset of pixels for which the gradient of \( u \) is non-zero, i.e., \( \{ r \in x | \nabla_r u \neq 0 \} \). For simplicity, we also use the phrase “receptive field of a convolutional network” to refer to “receptive field of a particular feature of a convolutional network”.

**B. Attack Formulation**

**Attack objective.** We focus on evasion attacks against an image classification model. Given a deep learning model \( \mathcal{M} \), an image \( x \), and its true class label \( y \), the goal of the attacker is to find an image \( x' \in \mathcal{C}(x) \subset \mathcal{X} \) satisfying a constraint \( \mathcal{C} \) such that \( \mathcal{M}(x') \neq y \). The constraint \( \mathcal{C} \) is defined by the attacker’s threat model, which we will describe below. This attack objective of inducing misclassification in any class apart from the true one is referred to as an untargeted attack. In contrast, if the attacker aims to misclassify the image to a particular target class \( y' \neq y \), it is called a targeted attack. We use the most effective white-box untargeted attack LaVAN [14] to evaluate the accuracy of undefended models while our provable defense results hold for any attack.

**Attacker capability:** For the adversarial patch attack, the attacker can arbitrarily modify pixels within a restricted region. This restricted region can be anywhere on the image, even over the salient object, but we assume there is only one such contiguous region of a fixed size. We note that this is the strongest threat model used in the existing literature on certified defenses against adversarial patches [18], [21], [22]. Formally, we use a binary image block \( p \in \mathbb{P} \subseteq \{0, 1\}^{W \times H} \) to represent the restricted region, where the pixels within the region are set to one. Then the constraint set \( \mathcal{C}(x) \) can be expressed as \( \{ x' = (1 - p) \odot x + p \odot x'' | x, x', x'' \in \mathcal{X}, x'' \in [0, 1]^{W \times H}, p \in \mathbb{P}, p \} \), where \( \odot \) refers to the element-wise product operator, and \( x'' \) is the content of the adversarial patch.

**Attacker knowledge:** A white-box adversary knows the model architecture and model weights while a black-box adversary can only query the model and get its output. We focus on the strong white-box adversary in this paper. An adversary can be also categorized as non-adaptive or adaptive based on its knowledge of the defense mechanism. Previous works show that most empirical defenses are fragile if the adversary knows the defense mechanism and that it is important to design and evaluate defenses against an adaptive attacker [10]–[12], [18]. Therefore, we consider the strongest adaptive attacker who knows the defense algorithms and parameters and provide a provable analysis for our defenses.

**Attack algorithm:** The adversarial patch attack can be formulated as a optimization problem as in most of the related literature [13], [14]. Here we abuse the notation to let \( \mathcal{M}(x') \) to be the final predicted confidence vector of the model, and \( y \) to be the one-hot encoded vector of the class.

\[
x' = \arg \max_{x' \in \mathcal{C}(x)} L((\mathcal{M}(x'), y))
\]

Here \( L(\cdot) \) refers to the cross-entropy loss. For the targeted attack with targeted class \( y' \neq y \), the optimization objective is slightly different as:

\[
x' = \arg \min_{x' \in \mathcal{C}(x)} L((\mathcal{M}(x'), y')
\]

Since we have \( x' = (1 - p) \odot x + p \odot x'' \), and \( p, x \) are fixed, the optimization is over \( x'' \), which is distinguished from conventional \( L_p \) adversary optimization [4], [5]. This optimization problem can be approximately solved with gradient-based optimization algorithms such as Stochastic Gradient Descent [29], [30].

**III. PatchGuard**

In this section, we first motivate the use of small receptive fields and robust masking in our defense. Towards this end, we analyze the behavior of the adversarial patch attack on ResNet-50 whose large receptive fields and insecure aggregation lead to high misclassification rates from even small adversarial patches. Next, we will give an overview of our general defense framework called PatchGuard, followed by our implementation of small receptive fields and details of our robust masking defense.

**A. Why are adversarial patches effective?**

Previous works [13], [14] on adversarial patches, surprisingly, show that model prediction can be manipulated by patches that occupy a very small portion of input images. For example, when using the LaVAN attack [14], a patch comprising of only 2% of the pixels of an image from the ImageNet dataset [24] is enough to achieve a nearly 100% untargeted attack success rate. In this subsection, we provide a case study for ResNet-50 [23] trained on ImageNet [24] and identify two critical reasons for the model vulnerability.

**Experiment I.** Applying the feature extractor \( F(\cdot) \) of ResNet-50 to an input image \( x \) gives us the local logits tensor in the shape of \( 7 \times 7 \times 1000 \), where \( 7 \times 7 \) is the number of local logits in the ResNet-50 architecture, and \( 1000 \) is the number of classes. Our first experiment aims to analyze the extent to which an adversarial patch can affect each of the \( 7 \times 7 \) local logits tensor \( \hat{u} \in \mathbb{R}^{1000} \). To do this, we modify the attack optimization objective to be \( \arg \max L(\text{softmax}(\hat{u}), y) \). We attack each \( \hat{u} \) individually and record the corresponding loss value which demonstrates how greatly the adversarial patch can influence the local logits. We randomly select 1000 images from the validation set for this experiment.
Vulnerability I: the small adversarial patch appears in the large receptive fields of all local features and easily manipulates each local prediction. We have two important observations from Experiment I. First, for 99.3% of images, all of their $7 \times 7$ local logits can be corrupted into a prediction for a wrong class. Each local feature of ResNet-50 is looking at a 483x483 pixel region in the input space [31]; therefore, even if the adversarial patch only appears in a restricted area, it is still within the receptive field of all local features and can easily manipulate the local predictions. Second, despite the powerful adversarial patch, we find that rare cases exist in which some of the local features are robust against the adversarial attack and retain their correct local predictions. We visualize two examples in Figure 2. The left column is the original image, the middle column demonstrates the location of the adversarial patch, and the right column is the $7 \times 7$ heatmap for the local logits attack. Each heatmap cell represents the normalized loss value for the attacked local logits; a brighter cell has a larger loss value and indicates a more effective attack. We can see the robust local logits (i.e., dark red cells) tend to be far away from the adversarial patch. The reason for this is that each local feature focuses exponentially more on the center of its receptive field, which is caused by the partial overlapping of the convolutional kernels in each layer. When the center of the receptive field is far away from the adversarial patch, the ability of the adversarial patch to influence the feature greatly decreases. These two observations motivate the use of small receptive fields: if the receptive field is small enough such that only a limited number of local features can be corrupted by the adversarial patch, we can provide a foundation for the robust prediction.

Experiment II. Our second experiment explores the distribution difference of local logits values, or class evidence, between clean and adversarial images. We use the untargeted LaVAN attack introduced in Section II-B to generate adversarial patches for 1000 random images from the ImageNet validation set. For each clean or adversarial image, we get the largest value in each of $7 \times 7$ local logits in $u = \mathcal{F}(x)$ and plot the histogram in Figure 3.

Vulnerability II: the small adversarial patch creates extremely large malicious local feature values and makes the average-based feature aggregation insecure. As we can see from Figure 3, the adversarial patch tends to create extremely large malicious class evidence to increase the chance of a successful attack. Conventional convolutional networks use average pooling to aggregate all local features, and thus are vulnerable to such large malicious feature values. This observation motivates our robust masking defense for a secure feature aggregation.

B. Overview of PatchGuard

In Section III-A, we identified the large receptive field and insecure aggregation of convolutional networks as two major sources of model vulnerability. In this subsection, we provide an overview of our defense that tackles both these problems.

Our goal is to propose a defense model $\mathcal{D}$ such that $\mathcal{D}(x) = \mathcal{D}(x') = y$ for any clean data point $(x, y) \in \mathcal{X} \times \mathcal{Y}$ and adversarial example $x' \in \mathcal{C}(x)$. $\mathcal{C}(x)$ is the constraint of adversary introduced in Section II-B. Note that the goal of our defense is to recover the correct prediction, which is harder than merely detecting the adversarial attack.

Figure 1 provides an overview of our defense framework. We consider a convolutional network $\mathcal{M}$ with small receptive fields. The feature extractor $\mathcal{F}(x)$ produces feature vectors $u$ extracted from each small region of the input image $x$. The concept of feature is general and can be either model prediction, confidence, logits, or feature map. Our defense framework is compatible with any convolutional network with small receptive fields, and we will present two instances of such networks in Section III-C. The small receptive field ensures that only a small fraction of features are corrupted by a localized adversarial patch. However, the insecure aggregation of these features via average pooling or sum might still result in misclassification $\mathcal{M}(u) = y' \neq y$. To address this vulnerability, we propose a robust masking algorithm for secure feature aggregation.

In robust masking, we detect and mask the corrupted features after getting the local feature tensor $u = \mathcal{F}(x)$. Since the number of corrupted local features is limited due to the small receptive field, the adversary has to create large feature values to dominate the global prediction. These large feature values lead to a distinct pattern and enables our detection of corrupted features. Moreover, our empirical analysis shows that model predictions on clean images are invariant to the removal of
partial features. Therefore, once the corrupted features are masked, we are likely to recover the correct prediction with $D(u) = y$ from the remaining local features, as shown in the right part of Figure 1. This defense introduces a trade-off for the attacker between increasing the adversarial behavior and being detected by our method leading to the removal of corrupted features. Such a trade-off enables provable robustness. We will introduce the defense details in Section III-D and its provable security analysis in Section IV.

C. Convolutional Networks with Small Receptive Fields

As defined in Section II, the receptive field of a convolutional network is a subset of image pixels that affect a particular local feature $\hat{u}$. Our defense framework is compatible with any network with a small receptive field. In this subsection, we present two instances of such networks used in this paper.

The first way to obtain a convolutional network with small receptive fields is to modify existing modern model architectures by replacing part of their large convolution kernels with $1 \times 1$ kernels. $1 \times 1$ kernels do not increase the size of the receptive field, and by controlling the fraction of $1 \times 1$ kernels we can obtain networks with tunable sizes of receptive fields. The BagNet architecture [25] uses this idea to create interpretable models while we use this model for provable robustness against adversarial patch attacks. In our experiments, we use the BagNet architecture adapted from ResNet-50.

The second approach is to train a model that only takes small pixel patches as inputs and aggregates predictions from all possible pixel patches to output the final prediction. In our notation, $F(x)$ will be the concatenation of features from all small pixel patches, and $M(F(x))$ aggregates these features from the small pixel patches for the final prediction. A similar architecture is used by the De-randomized Smoothing defense (DS) [22] where the model aggregates predictions on all possible ablated images, which are small pixel patches in the shape of square or rectangle. DS aggregates prediction counts from small pixel patches to derive provable robustness while we use robust masking for a secure aggregation with higher provable robust accuracy. In our experiments, we train a ResNet classifier whose input is a rectangular pixel band.

D. Robust Masking

Given that an adversarial patch can only corrupt a limited number of local features with small receptive fields, it is likely to result in a small region of abnormally high feature values to induce misclassification. We aim to detect and mask this corrupted region so the final classification is not influenced by these adversarial feature values. We will then recover the correct prediction as long as clean images are robust to masks, which we demonstrate is indeed the case for the state-of-the-art models we consider. The defense algorithm is shown in Algorithm 1. It takes a clean or adversarial image as its input and aims to output the correct prediction.

Clipping. As shown in Algorithm 1, the first step of our defense is to get the class prediction $\bar{y}$ and its corresponding clipped local feature tensor $\hat{y}$ as the preparation for malicious region detection. The clipping values $c_l, c_h$ depend on the type of feature we are using. If we use logits, we will set $c_l = 0, c_h = \infty$ by default. We clip the negative values to zero since they make little contribution to the correct prediction of clean images but can be abused by the adversary to reduce the class evidence of the true class. If the feature refers to confidence tensor or one-hot encoded prediction, the clipping is optional, or equivalently $c_l = 0, c_h = 1$, since the range of the feature values are already bounded.

Detection. We use the subprocedure DETECT to examine the clipped local feature tensor $\hat{u}_y$ and detect the malicious region. DETECT takes feature tensor $\hat{u}_y$, normalized detection threshold $T \in [0, 1]$, and a set of sliding windows $W$ as inputs. A window is a binary mask in the feature space whose size is determined by the number of local features that can be corrupted by the adversarial patch. Formally, we can computed the window size as $\text{window size} = \left\lceil \frac{\text{patch size} + \text{receptive field size} - 1}{\text{stride}} \right\rceil$, where stride is the pixel distance between two adjacent receptive centers. We represent each window $w$ with a binary feature map in $\{0, 1\}^{W' \times H'}$, where pixels within the window have values of one. This representation is essentially the same as the binary image mask mentioned in Section II-B except that window is for local feature tensor and image block is for input image tensor. We use different terms to avoid potential confusion. To detect the malicious region, DETECT calculates the sum of feature values, or class evidence for class $\bar{y}$, within

```
Algorithm 1 Robust masking

Input: image $x$, model $M$, local feature extractor $F$, clipping range $[c_l, c_h]$, the set of sliding windows $W$, and detection threshold $T$.

Output: robust prediction $\bar{y}$

1: procedure ROBUSTMASKING
2:    $\bar{y} \leftarrow M(x)$ \textcolor{red}{Model prediction $\bar{y}$}
3:    $u_y \leftarrow F(x, \bar{y})$ \textcolor{red}{Local feature tensor for class $\bar{y}$}
4:    $\hat{u}_y \leftarrow \text{CLIP}(u_y, c_l, c_h)$ \textcolor{red}{Clipped local features $\hat{u}_y$}
5:    $w^* \leftarrow \text{DETECT}(\hat{u}_y, T, W)$ \textcolor{red}{Suspicous window $w^*$}
6:    if $w^* \not\perp$ then
7:        $\bar{y} \leftarrow M(F(x) \odot (1 - w^*))$ \textcolor{red}{Masked prediction}
8:    else
9:        $\bar{y} \leftarrow \hat{y}$ \textcolor{red}{Retained original prediction}
10: end if
11: return $\bar{y}$
12: end procedure

13: procedure DETECT($\hat{u}_y, T, W$)
14:    $w^* \leftarrow \arg \max_{w \in W} \text{SUM}(w \odot \hat{u}_y)$ \textcolor{red}{Detection}
15:    $t \leftarrow \text{SUM}(w^* \odot \hat{u}_y)/\text{SUM}(\hat{u}_y)$ \textcolor{red}{Normalization}
16:    if $t \leq T$ then
17:        $w^* \leftarrow \perp$ \textcolor{red}{Return $\perp$ if not exceeding threshold $T$}
18:    end if
19:    return $w^*$
20: end procedure
```
every possible window and identifies the window with the highest sum of class evidence as the suspicious window. If the normalized highest class evidence exceeds the threshold $T$, we return the corresponding window $w^*$; otherwise, we return $\bot$.

**Masking.** If we detect a suspicious area in the local feature space, we remove the suspicious features and do prediction on the masked features to get $y = \mathcal{M}(\mathcal{L}(x) \odot (1 - w^*))$. Our empirical analysis shows that clean image predictions are usually invariant to such feature masking; therefore, this masking operation provides us with a robust classification model rather than a simple detection defense. Similarly, our defense is able to retain high clean accuracy, because even if we incorrectly detect and removes suspicious features of a clean image, the model is still able to make a correct prediction with high probability.

**IV. PROVABLE ROBUSTNESS ANALYSIS**

In this section, we provide the provable robustness analysis for our robust masking defense. For any clean image $x$ and a given model $\mathcal{M}$, we will determine whether the attacker can bypass the robust masking defense, *even with knowledge of the defense*. Recall that our threat model only allows the adversarial patches to be within a fixed-size region, where the size of the region is a tunable security parameter. Given this threat model, we know that all the corrupted features will also be within a small window in the feature map space when using a model with a small receptive field, with the size of this window being determined by the patch size and the size of the model's receptive field.

**Trade-off for the attacker.** With the robust masking defense, we put the adversary in a dilemma. If the adversary wants to succeed in the attack, it needs to increase the class evidence of a wrong class. However, this increase of class evidence will trigger our detection and masking mechanism. If the required minimum increase in malicious class evidence is large enough to trigger the detection, the adversary has little chance to succeed. We provide the pseudocode for the provable robustness analysis in Algorithm 2. Next, we will explain Algorithm 2 and then prove that our approach provides provable robustness in Theorem 1.

**Effect of clipping.** To start with, we return False if the clean image $x$ cannot be correctly classified. Next, we iterate over all possible target classes to derive provable robustness for the untargeted attack. For each target class, we first get the clipped local feature for the true class and the target class. We always clip negative feature values to zero and leave positive values unchanged. If the feature is confidence tensor or prediction tensor, the feature values are already positive only. If the feature is logits, negative values do not contribute much to the clean image prediction but can be abused by the attacker to reduce the class evidence of the true class. Putting this constraint improves our robust accuracy.

**Analyzing masking.** The main analysis is in the inner for loop. We examine every possible window for the feature tensor. For each possible malicious window $w$, we first ignore the elements within this window with the operation $(1 - w) \odot (-)$ and calculate the sum of the difference between two local feature tensors $\hat{u}_y, \hat{u}_{y'}$. The result $\delta$ is the minimum class evidence difference the adversary has to achieve by controlling the content within its malicious window $w$ for a successful...

---

### Algorithm 2: Analysis of provable accuracy for robust masking

**Input:** image $x$, true class $y$, label set $\mathcal{Y}$, model $\mathcal{M}$, local feature extractor $\mathcal{F}$, the set of sliding windows $\mathcal{W}$, and detection threshold $T$.

**Output:** Whether the image $x$ has provable robustness

1. **procedure** **ProvabilityAnalysisMasking**
2. $\hat{y} \leftarrow \mathcal{M}(x)$ $\triangleright$ Get the model prediction
3. **if** $\hat{y} \neq y$ **then**
4. **return** False $\triangleright$ Original prediction is incorrect
5. **end if**
6. **for** each $y' \in \mathcal{Y}$ and $y' \neq y$
7. **▷** Get clipped local feature for true and target class $\hat{u}_y \leftarrow \text{CLIP}(\mathcal{F}(x, y), 0, \infty)$
8. $\hat{u}_{y'} \leftarrow \text{CLIP}(\mathcal{F}(x, y'), 0, \infty)$
9. **for** each $w \in \mathcal{W}$
10. **▷** Required minimum class evidence change
11. $\delta \leftarrow \text{SUM}(((1 - w) \odot (\hat{u}_y - \hat{u}_{y'})))$
12. **Cases for a potential successful attack**
13. $s_1 \leftarrow \text{NoWindow}(\hat{u}_{y'}, w, \delta)$
14. $s_2 \leftarrow \text{BenignWindow}(\hat{u}_y, \hat{u}_{y'}, w, \mathcal{W})$
15. $s_3 \leftarrow \text{PartMalWindow}(\hat{u}_y, \hat{u}_{y'}, w, \mathcal{W})$
16. **if** $s_1 < T$ **or** $s_2 > \delta$ **or** $s_3 > \delta$ **or** $\delta < 0$ **then**
17. **return** False
18. **end if**
19. **end for**
20. **end procedure**

---

**Algorithm 2.1: Analysis of provable accuracy for robust masking**

1. **procedure** **ProvabilityAnalysisMasking**
2. $\hat{y} \leftarrow \mathcal{M}(x)$ $\triangleright$ Get the model prediction
3. **if** $\hat{y} \neq y$ **then**
4. **return** False $\triangleright$ Original prediction is incorrect
5. **end if**
6. **for** each $y' \in \mathcal{Y}$ and $y' \neq y$
7. **▷** Get clipped local feature for true and target class $\hat{u}_y \leftarrow \text{CLIP}(\mathcal{F}(x, y), 0, \infty)$
8. $\hat{u}_{y'} \leftarrow \text{CLIP}(\mathcal{F}(x, y'), 0, \infty)$
9. **for** each $w \in \mathcal{W}$
10. **▷** Required minimum class evidence change
11. $\delta \leftarrow \text{SUM}(((1 - w) \odot (\hat{u}_y - \hat{u}_{y'})))$
12. **Cases for a potential successful attack**
13. $s_1 \leftarrow \text{NoWindow}(\hat{u}_{y'}, w, \delta)$
14. $s_2 \leftarrow \text{BenignWindow}(\hat{u}_y, \hat{u}_{y'}, w, \mathcal{W})$
15. $s_3 \leftarrow \text{PartMalWindow}(\hat{u}_y, \hat{u}_{y'}, w, \mathcal{W})$
16. **if** $s_1 < T$ **or** $s_2 > \delta$ **or** $s_3 > \delta$ **or** $\delta < 0$ **then**
17. **return** False
18. **end if**
19. **end for**
20. **end procedure**
There are four possible cases for the output window \( w^* \) of the DETECT subprocedure, and we will discuss the condition for the potentially successful attack in each case.

1) **Case I: no suspicious window detected.** The attack succeeds only when the required minimum increase of the target class evidence \( \delta \) does not exceed the detection threshold. If this condition is satisfied (i.e., \( s_1 < T \)), the algorithm returns \text{False}.

2) **Case II: a benign window is incorrectly detected.** In this case, the detected window \( w^* \) satisfies \( \sum (w^* \odot w) = 0 \) (i.e., \( w^* \) does not overlap with malicious window \( w \)). We calculate the sum of target class evidence and true class evidence within the benign window \( w^* \) to be \( t \) and \( s_2 \), respectively. Since our defense will incorrectly zero out features within \( w^* \), the adversary achieves a difference of \( s_2 - t \). Moreover, the adversary can also increase the malicious class evidence in its malicious window \( w \) to \( t \), and total class evidence achieved is \( s_2 - t + t = s_2 \). If \( s_2 > \delta \), the attack might succeed, and the algorithm returns \text{False}. Note that the increase of malicious evidence within the malicious window \( w \) cannot be larger than \( t \); otherwise, the malicious window instead of the benign window will be detected.

3) **Case III: the malicious window is partially detected.** In this case, the detected window \( w^* \) partially overlaps with the malicious window \( w \), or \( \sum (w^* \odot w) > 0 \). Following a similar reasoning as in Case II, the adversary can achieve a difference of \( s = \sum (w^* \odot y^* \odot (1 - w)) \) for each possible partially overlapped window \( w^* \). We output the largest \( s \) as \( s_3 \) and return \text{False} if \( s_3 > \delta \).

4) **Case IV: the malicious window is perfectly detected.** In this case, we have \( w = w^* \). The attack can succeed only when feature masking will cause prediction change, or \( \delta < 0 \). Therefore, if the algorithm checks every possible \( w \) and possible target class \( y^* \) and returns \text{True} instead of \text{False}, the input image can never be attacked and has provable robustness.

### V. Evaluation

In this section, we provide a comprehensive evaluation of the proposed defense mechanism. We report the provable robustness of our defense on a 10-class subset of ImageNet (called ImageNet2 [27]), the entire 1000-class ImageNet [24], and CIFAR-10 [28] for various patch sizes. We instantiate our defense with multiple different convolutional networks with small receptive fields and compare the results with previous defenses [18], [21], [22]. We also provide a detailed analysis of our defense performance with different parameters.

#### A. Experiment Setup

**Datasets.** We report our main provable robustness results on ImageNet2 [27], ImageNet [24] and CIFAR-10 [28] datasets. ImageNet2 and ImageNet images are in high resolution and will be resized and cropped to 224x224 or 299x299 before being fed into different models while CIFAR-10 images have a lower resolution of 32x32. ImageNet2 is a 10-class subset of the large ImageNet dataset, we will provide a detailed analysis of different defense parameters on this smaller dataset. We include the details of these three datasets in the Appendix.

**Models.** We analyze the performance of five different models: ResNet-50, BagNet-33, BagNet-17, BagNet-9, and a de-normalized smoothed ResNet with band size of 25 (DS-25-ResNet). These 5 models have a similar network structure but have different receptive fields of 48x48, 33x33, 17x17, 9x9, and 25x299, respectively. The results from five different models demonstrate the effect of receptive field sizes on the defense. For ImageNet2, we use models [32]–[34] pre-trained on ImageNet and retrain the entire models. For ImageNet, we use the pre-trained models [32]–[34].

**Defenses.** We analyze the defense performance against a single adversarial patch of size 23x23, 31x31, and 39x39, which takes up around 1%, 2%, and 3% pixels of the 224x224 images. For CIFAR-10, we report results for a 2x2 patch and a 5x5 patch, which have 0.4% and 2.4% pixels of the 32x32 images. Our provable robustness results provide a bound against any adversarial patch attack, including an adaptive white-box attacker that tries all possible locations for an adversarial patch. We use an
empirical analysis to show the limitation of undefended vanilla models. We use the most effective non-universal white-box attack which is LaVAN [14]. We formulate the LaVAN attack as an untargeted one for its effectiveness. Due to computational constraints, we only consider a random location for each image in our empirical evaluation, which is a similar approach used in the LaVAN paper [14]. We empirically find that attacks with a random location are effective against undefended vanilla models.

**Defenses.** We report the defense performance for our robust masking defense with BagNet (Mask-BN) and with a de-randomized smoothed ResNet (Mask-DS) on ImageNette, ImageNet, and CIFAR-10. We also compare with the existing Clipped BagNet (CBN) [21]. De-randomized Smoothing (DS) [22] and Interval Bound Propagation based certified defense (IBP) [18]. We only report results for the IBP defense on CIFAR-10 because this defense is too computationally expensive to scale to ImageNette and ImageNet. The default parameters of our defense are listed in Table II. For previous defenses, we use the best parameters reported in the original papers.

**Determining the window size.** A crucial step of our provable robustness analysis is determining the size of sliding windows. As introduced in Section III-D the window size can be computed as \( \text{window size} = \left\lceil \frac{\text{patch size} + \text{receptive field size} - 1}{\text{stride}} \right\rceil \), where stride is the pixel distance between two adjacent receptive centers. For the network architectures [32], [33] used in this paper, we have stride = 8 for BagNet and stride = 1 for DS-ResNet.

### TABLE III
**CLEAN AND EMPIRICAL ADVERSARIAL ACCURACY FOR UNDEFENDED MODELS**

| Dataset | ImageNette | ImageNet | CIFAR-10 |
|---------|------------|----------|----------|
| Patch size | 1% pixels | 2% pixels | 3% pixels | 1% pixels | 2% pixels | 3% pixels | 0.4% pixels | 2.4% pixels |
| Accuracy | clean | adv. | clean | adv. | clean | adv. | clean | adv. | clean | adv. | clean | adv. | clean | adv. | clean | adv. | clean | adv. |
| ResNet | 99.6 | 47.3 | 99.6 | 14.9 | 96.6 | 4.0 | 76.1 | 11.7 | 76.1 | 2.8 | 76.1 | 0.2 | 91.4 | 83.4 | 91.4 | 38.2 |
| BagNet | 95.7 | 39.1 | 95.7 | 11.7 | 95.7 | 2.9 | 58.7 | 8.4 | 58.7 | 3.8 | 58.7 | 2.0 | 81.0 | 35.8 | 81.0 | 0.7 |

### TABLE IV
**CLEAN AND PROVABLE ROBUST ACCURACY FOR DIFFERENT DEFENSES**

| Dataset | ImageNette | ImageNet | CIFAR-10 |
|---------|------------|----------|----------|
| Patch size | 1% pixels | 2% pixels | 3% pixels | 1% pixels | 2% pixels | 3% pixels | 0.4% pixels | 2.4% pixels |
| Accuracy | clean | adv. | clean | adv. | clean | adv. | clean | adv. | clean | adv. | clean | adv. |
| Mask-BN | 93.0 | 87.6 | 92.3 | 85.2 | 91.6 | 81.6 | 58.5 | 21.4 | 57.4 | 14.4 | 57.1 | 10.9 | 73.2 | 56.0 | 70.4 | 37.7 |
| Mask-DS | 91.3 | 83.1 | 90.3 | 80.2 | 89.8 | 77.7 | 39.1 | 21.3 | 37.6 | 17.7 | 36.4 | 14.8 | 82.5 | 71.9 | 80.3 | 61.3 |
| IBP [18] | 94.9 | 74.5 | 94.9 | 60.8 | 94.9 | 45.9 | 49.5 | 13.4 | 49.5 | 7.0 | 49.5 | 3.1 | 75.2 | 48.1 | 75.2 | 14.7 |
| CBN [19] | 92.0 | 82.3 | 92.0 | 79.1 | 92.0 | 75.1 | 44.3 | 17.6 | 44.3 | 14.0 | 44.3 | 11.1 | 83.9 | 68.8 | 83.9 | 57.8 |

**B. Provable Robustness Results**

In this subsection, we report our main evaluation results for provable robustness. **Clean accuracy** refers to the classification accuracy on the clean validation images, and **adversarial/robust accuracy** refers to the model accuracy on adversarial images from the validation set.

**Vanilla models are vulnerable to adversarial patch attacks.** We report the clean and empirical adversarial accuracies of undefended vanilla ResNet and BagNet in Table III. We attack these models with the untargeted LaVAN attack at one random location for each image. We report the empirical attack results for the entire validation set for ImageNette and CIFAR-10 as well as 2000 randomly selected validation images from ImageNet. We can see both vanilla ResNet and BagNet are vulnerable to adversarial patch attacks; a small patch consisting of only 3% pixels can degrade the model accuracy close to zero on ImageNet. Therefore, it is necessary to defend against such attacks.

**Robust Masking significantly improves model robustness with a high clean accuracy.** We report our main provable robustness results, against all possible adversaries, in Table IV. Our results for Mask-BN and Mask-DS demonstrate that our defense leads to high provable robust accuracy across datasets and models. For a 2% pixel patch, Mask-BN achieves an 85.2% provable accuracy on ImageNette and 14.4% on ImageNet while Mask-DS has an 81.0% provable accuracy on ImageNet and 17.7% on ImageNet. For a 2.4% pixel patch on CIFAR-10, Mask-BN and Mask-DS achieve a provable robust accuracy of 37.7% and 61.3%, respectively.

Simultaneously, our defenses retain high clean accuracy. For a 2% pixel patch, Mask-BN has a 92.3% clean accuracy on ImageNette and 57.4% on ImageNet while Mask-DS achieves a 90.3% clean accuracy on ImageNette and 37.6% on ImageNet. For CIFAR-10, Mask-BN and Mask-DS have a high
clean accuracy of 75.8% and 80.3%. We note that the drop in clean accuracy is moderate compared to undefended models. For ImageNette, the clean accuracy drop of Mask-BN and Mask-DS compared with undefended ResNet is within 10%. For Mask-BN, the accuracy drop from the undefended BagNet is only around 3%. We can also see a similar moderate clean accuracy drop for the other two datasets. Robust masking uses a different window size for different patch sizes, and therefore the clean accuracy for different patches varies slightly.

**Robust Masking achieves higher provable accuracy than all previous defenses.** We compare our defense performance with existing defenses across three datasets. On ImageNette, our Mask-BN achieves the best provable accuracy for three different patch sizes. Its clean accuracy is slightly lower than CBN, but we note that our provable robust accuracy is much higher, especially when the patch size is large. When the patch takes up of 3% image pixels, the provable accuracy of mask-BN is 35.7% higher than that of CBN while the clean accuracy is only 3.3% lower. Our mask-BN also outperforms DS in terms of clean accuracy and provable accuracy. Mask-DS has better provable accuracy but lower clean accuracy compared with DS and CBN. We do not report results for IBP on ImageNette and ImageNet because IBP is too computationally expensive to scale to high-resolution images.

On ImageNet, our defenses also achieve the best performance. Both Mask-BN and Mask-DS have a much higher provable accuracy than CBN. The provable robustness of Mask-BN and Mask-DS is either comparable or moderately higher than DS but we achieve significantly higher clean accuracy (57.1% for Mask-BN at 3% pixels, compared to 44.3% for DS).

For CIFAR-10, both Mask-BN and Mask-DS defenses significantly outperform IBP in terms of clean accuracy and provable accuracy. Our Mask-DS achieves the best provable robustness compared to all prior work including CBN and DS, but its clean accuracy is slightly lower than DS. Mask-BN does not perform as well on CIFAR-10 because we find that lower resolution images in CIFAR-10 affect the performance of the BagNet architecture. We expect that the performance of the approach will improve with better architectures. We note that CBN relies only on BagNet as opposed to our approach of robust masking. CBN is thus very fragile on CIFAR-10 due to the lower resolution images, achieving the worst provable performance in prior works.

**Takeaways.** Our evaluation shows the effectiveness of our proposed defenses, achieving state-of-the-art provable robustness on all three datasets. We find that BagNet-based defenses (Mask-BN and CBN) perform well on ImageNette and ImageNet but are fragile on CIFAR-10 due to its low image resolution. Meanwhile, De-randomized Smoothing based defenses (Mask-DS and DS) perform better on CIFAR-10. This shows that while the Robust Masking defense always improves 2The provable robust accuracy for 2% pixel patches reported in DS paper [23] is 14.5%, but that result is for a subset of 1000 images. We successfully reproduced their results of 14.5% and scaled the evaluation to the entire validation set, which resulted in a provable accuracy of 14.0%.

| Window size | 0x0 | 2x2 | 4x4 | 6x6 | 8x8 |
|-------------|-----|-----|-----|-----|-----|
| Masked accuracy | 95.7% | 95.7% | 95.6% | 95.5% | 95.3% |
| % images      | 4.3% | 5.3% | 6.6% | 8.0% | 9.8% |
| % windows per image | 0% | 0.05% | 0.2% | 0.4% | 0.7% |

**TABLE VI INvariance of BagNet-17 predictions to feature masking**

| Clean | 1% pixels adv. | 2% pixels adv. | 3% pixels adv. |
|-------|----------------|----------------|----------------|
| ResNet-50 | 99.6% | 47.3% | 14.9% | 4.0% |
| BagNet-33 | 97.3% | 27.0% | 4.8% | 1.1% |
| BagNet-17 | 95.7% | 39.1% | 11.7% | 2.9% |
| BagNet-9 | 92.7% | 26.8% | 14.8% | 3.8% |

**TABLE V ACCURACY OF VANILLA BAGNETS**

**C. Detailed Analysis of Robust Masking**

In this subsection, we use ImageNette to analyze the performance of robust masking under different defense parameters. We will only report results for Mask-BN when the observations from Mask-BN and Mask-DS are very similar.

**Undefended vanilla models.** We report the clean and adversarial accuracy of vanilla models with different receptive field sizes in Table [V]. The clean accuracy of BagNet decreases as the receptive field of BagNet becomes smaller since each local feature receives less information. Furthermore, we observe that all undefended vanilla models are vulnerable to adversarial patch attacks regardless of their receptive field sizes; a patch consisting of only 3% pixels can achieve an attack success rate higher than 96% across all models, which makes our defense necessary.

**Prediction invariance of vanilla models to feature masking.** In our robust masking defense, we detect and mask corrupted features. If the model can make correct predictions from the aggregation of remaining features, we can recover the correct prediction. Therefore, we first analyze the prediction invariance of vanilla models to partial feature masking. We take BagNet-17, which has 26 × 26 local features, for the case study. We mask out features within a set of sliding windows of different sizes and record the prediction from the remaining features. We report the averaged accuracy for all possible masked feature (masked accuracy), the percentage of images for which at least one masked prediction is incorrect (% images), and the percentage of masks that will cause prediction change for each image (% windows per image). As shown in Table [VI] the overall averaged masked accuracy is high, and the percentage of images and windows for which the prediction changes is low. Note that when there is at least one masked window that could cause a prediction change, the adversary can simply put a patch at the corresponding region of the input image and have a trivially successful untargeted attack. The small fraction of images with prediction changes enables us to achieve a high provable robustness while
maintaining clean accuracy by recovering correct predictions. Moreover, as the size of window increases from 0x0 to 8x8, the averaged accuracy of masked feature falls only a little; the percentage of images that have at least one prediction change and the fraction of windows that cause prediction changes only increase slightly. We find that similar observations hold for ResNet and DS-ResNet.

Effect of clipping on vanilla models. The default clipping values for Mask-BN are \( c_l = 0, c_h = \infty \) when the feature type is the logits tensor. We vary the clipping value for the local logits for ResNet and BagNet to determine how the clean accuracy changes, and the results are shown in Table VII. We find that clipping the negative values only slightly affects the clean accuracy. When we decrease the positive clipping value \( c_h \), the clean accuracy of the model also decreases. We notice that models with smaller receptive fields are more sensitive to clipping. This is because models with small receptive fields only have a small number of correct local predictions. The corresponding correctly predicted local logits have to use large logit values to dominate the global prediction, which leads to the sensitivity to clipping. As shown in Figure 3, the logits of the adversarial images tend to have large values. If we set \( c_h \) to the largest clean logit value, we will not affect the clean accuracy and can improve the empirical robustness against the adversarial patch. We note that setting \( c_h \) to a large real number will not affect the provable robustness compared with \( c_h = \infty \). When features correspond to confidence or prediction values, we do not suggest any additional clipping since those values are already bounded between 0 and 1.

Effect of receptive field sizes on defended models. We report clean accuracy and provable robust accuracy of our defense for BagNet-33, BagNet-17, and BagNet-9, which have a receptive field of 33x33, 17x17, and 9x9, respectively, against different patch sizes in Table VIII. As shown in the table, a model with a larger receptive field has better clean accuracy. However, a larger receptive field results in a larger fraction of corrupted features and thus a larger gap between clean accuracy and provable robust accuracy. We can see that though BagNet-33 has a higher clean accuracy than BagNet-17, its gap between clean accuracy and provable robust accuracy is larger, which results in a similar or slightly poorer provable robust accuracy compared with BagNet-17. Our evaluation shows that BagNet-17 achieves the best trade-off between clean and robust accuracy.

Effect of using different feature types for defended models. In this analysis, we study the performance of the robust masking defense when using different types of features, namely logits, confidence values, and predictions. The results for BagNet-17 with different features are reported in Table IX. As shown in the table, using logits as the feature type has a much better performance than confidence and prediction in terms of clean accuracy and provable accuracy. The main reason for this observation is that BagNet is trained with logits aggregation. Our additional analysis shows that BagNet does not have high model performance when trained with confidence or prediction aggregation; therefore, we use logits as our default feature type for Mask-BN. Interestingly, Mask-DS exhibits a different behavior. As shown in Table X, Mask-DS works better when we use prediction or confidence as feature types due to its different training objectives. In conclusion, the performance of different feature types largely depends on the training objective of the network with small receptive fields, and should be appropriately optimized to determine the best defense setting.

Effect of the detection threshold on defended models. We study the model performance of BagNet-17 against a 2% pixel patch as we change the detection threshold \( T \) from 0.0 to 1.0. A threshold of zero means our detection will always return a suspicious window even the input is a clean image while a threshold of one means no detection at all. We report the clean accuracy, provable robust accuracy, and false positive (FP) rates for detection of suspicious windows on clean images in the left part of Table XI. When we increase the detection threshold \( T \), we reduce the false positive rates for detection of a suspicious window in clean images, at the cost of making it easier for an adversarial patch to succeed via Case I (no

| Patch size | 1% pixels | 2% pixels | 3% pixels |
|------------|-----------|-----------|-----------|
|            | clean     | adv.      | clean     | adv.      | clean     | adv.      |
| BagNet-33  | 94.6%     | 87.7%     | 93.9%     | 84.4%     | 93.0%     | 79.6%     |
| BagNet-17  | 93.0%     | 87.6%     | 92.3%     | 85.2%     | 91.6%     | 81.6%     |
| BagNet-9   | 88.0%     | 82.7%     | 87.3%     | 80.5%     | 86.4%     | 77.4%     |

TABLE VIII

| Patch size | 1% pixels | 2% pixels | 3% pixels |
|------------|-----------|-----------|-----------|
|            | clean     | adv.      | clean     | adv.      | clean     | adv.      |
| Logits     | 91.5%     | 77.9%     | 91.8%     | 70.4%     | 90.5%     | 63.7%     |
| Confidence | 91.1%     | 83.6%     | 90.7%     | 81.0%     | 90.1%     | 78.3%     |
| Prediction | 91.3%     | 83.1%     | 90.3%     | 80.2%     | 89.8%     | 77.7%     |

TABLE IX

| Patch size | 1% pixels | 2% pixels | 3% pixels |
|------------|-----------|-----------|-----------|
|            | clean     | adv.      | clean     | adv.      | clean     | adv.      |
| Logits     | 93.0%     | 87.6%     | 92.3%     | 85.2%     | 91.6%     | 81.6%     |
| Confidence | 84.5%     | 77.8%     | 84.1%     | 75.1%     | 83.3%     | 71.8%     |
| Prediction | 80.3%     | 72.4%     | 80.2%     | 69.6%     | 79.6%     | 66.5%     |

TABLE X
### TABLE XI
PROVABLE ACCURACY AND FAILURE CASE BREAKDOWN OF BAGNET-17 WITH DIFFERENT DETECTION THRESHOLD

|        | Clean | Provable Accuracy | FP | Incorrect | Case I | Case II | Case III | Case IV |
|--------|-------|-------------------|----|-----------|--------|---------|----------|---------|
| T-0.0  | 92.3% | 85.2%             | 100% | 4.8%      | 0%     | 2.8%    | 6.7%     | 0.2%    |
| T-0.2  | 92.6% | 82.9%             | 39.9% | 4.8%      | 8.3%   | 0.4%    | 3.1%     | 0.2%    |
| T-0.4  | 94.2% | 69.2%             | 3.2% | 4.8%      | 25.0%  | 0%      | 0.5%     | 0.2%    |
| T-0.6  | 94.9% | 42.3%             | 0.3% | 4.8%      | 52.4%  | 0%      | 0%       | 0.2%    |
| T-0.8  | 95.1% | 10.4%             | 0%  | 4.8%      | 84.4%  | 0%      | 0%       | 0.2%    |
| T-1.0  | 95.1% | 0%                | 0%  | 4.8%      | 94.8%  | 0%      | 0%       | 0.2%    |

suspicious window detected). However, we note that FP in the detection phase for clean images have minimal impact on the clean accuracy because our models are invariant to feature masking, as already shown in Table [VI]. Thus, we find $T = 0$ to be the best choice for this dataset; it results in the highest provable robust accuracy of 85.2% while only incurring a 2.8% clean accuracy drop compared to $T = 1$.

**Breakdown of provable analysis on defended models.** We study the failure cases of our provable analysis and report the results in the right part of Table [XI]. Note that each image might be attacked via multiple failure cases introduced in Section [IV]. We only count the first found failure case; if we find an image can be attacked via Case I, we will not consider whether it can be attacked via Case II, III, or IV. As shown in Table [XI] when we set the detection threshold to zero, the majority of failure cases come from the original prediction being incorrect (Incorrect: 4.8%), an incorrectly detected benign window (Case II: 2.8%), and a partially detected malicious window (Case III: 6.7%). We conclude that the high vanilla model accuracy and model invariance to feature masking are the keys to further improve our provably robust defense. Note that the case of no detected suspicious window (Case I) is 0% because we always output a detected window with $T = 0$; the case of a perfectly detected malicious window (Case IV) is nearly 0% because most of these failure instances are counted in Case III.

**VI. DISCUSSION**

In this section, we will show that our defense is a generalization of previous provable defenses and discuss future research directions pertaining to our defense.

**A. PatchGuard as a Generalization of Previous Provable Defenses**

In this subsection, we will show that our defense framework is a generalization of the Clipped BagNet (CBN) [21] and De-randomized Smoothing defenses (DS) [22].

**Clipped BagNet (CBN).** CBN [21] proposes clipping the local logits tensor with function $CLIP(u) = \tanh (0.05 \cdot u - 1)$ to improve the robustness of BagNet [25]. Since the range of $\tanh(\cdot)$ is bounded by $(-1, 1)$, the adversary can achieve at most $2 \times k$ difference in clipped logits values between the true class and any other class, where $k$ is the number of corrupted local logits due to the adversarial patch. In its provable robustness analysis, CBN calculates the difference between sum of logits values for the predicted class and the second predicted class as $\delta$; if $\delta > 2 \times k$, CBN certifies the robustness of the input clean image. To reduce our Mask-BN defense to CBN, we can set our feature type to logits, the detection threshold to $T = 1$, and adjust the clipping values $c_l$ and $c_r$ or the clipping function $CLIP(\cdot)$. Our evaluation shows that our defense significantly outperforms CBN across three different datasets. The major reason for this performance difference is the use of sub-optimal clipping function in CBN (i.e., $\tanh(\cdot)$). Our analysis shows that the negative values contribute little to the prediction but can be abused by the adversary; thus, this clipping design leads to a loose bound.

**De-randomized Smoothing (DS).** DS [22] trains a ‘smoothed’ classifier on ablated images (which are pixel patches obtained from the original images) and computes the predicted class as the class with the largest count in local predictions made from all ablated images. The provable robustness analysis of DS only considers the largest and second-largest counts of local predictions. If the gap between the two largest counts is larger than two times the upper bound of the number of corrupted predictions, DS certifies the robustness of the image. When we set the feature type to prediction vector and only consider the worst case in which our DETECT in Algorithm [I] always falsely outputs a benign window, our Mask-DS can be reduced to DS. The certification process of DS discards the spatial information of each prediction and leads to a looser bound. In contrast, our robust masking utilizes the spatial information that all corrupted features are within a small window in the feature space. Only when the original predictions within the malicious window and the falsely detected benign windows all vote for the true class, will we reach the extreme bound used in DS. This extreme case is unlikely to happen in our provable analysis of robust masking. Therefore, our robust masking can derive a tighter bound than DS, and our evaluation results in Section [V] also support this claim.

**B. Future Work**

**Improve the design of networks with small receptive fields.** The foundation of our defense is convolutional networks with a small receptive field, and we use two such instances via BagNet [25] and DS-ResNet [22]. Our current defense performance is largely constrained by the clean accuracy of BagNet and DS-ResNet, despite the fact that our defense framework is compatible with any network with small receptive fields. We hope that our work inspires the design of novel network architectures that have a small receptive field.
while maintaining state-of-the-art clean accuracy, which would further boost the performance of our defense. 

Use a “soft” constraint via quantitatively bounding the contribution of each pixel. In our work, we use a hard constraint to limit the number of corrupted features due to an adversarial patch to provide provable robustness. However, as discussed in Section [III-A], each local feature focuses exponentially more on the center of its receptive field, and pixels far away from the center of the receptive field only have a limited influence on the local feature. This behavior can be regarded as a “soft” limitation of the receptive field size. If we can quantitatively bound the influence of each pixel on the local feature, we can have a more fine-grained bound on the robust accuracy. We hope that our work inspires future research on the relationship between receptive field sizes and model robustness.

Explore alternative secure feature aggregation approaches. We present robust masking to compute robust predictions from partially corrupted features, and we note that it is just a single instance of our broader defense framework. Our defense framework turns the problem of designing an adversarial patch defense into a robust aggregation problem, i.e., how can we make a robust prediction from a partially corrupted feature tensor? Techniques from robust statistics such as clipping, median, truncated mean as well as differential privacy [35] can also be incorporated in our framework. We find that robust masking significantly outperforms simple aggregation methods like clipping and median, but our approach is compatible with alternative aggregation techniques.

Extend defenses to tasks other than image classification. Our defense design and evaluation are focused on the white-box untargeted attacks against image classification models. An interesting direction for future work would be to use our framework to defend against adversarial patch attacks for other tasks like object detection [36], [37].

VII. Related Work

A. Localized Adversarial Perturbations

Most adversarial example research focuses on $L_p$-norm bounded perturbations added to the entire input. In contrast, localized adversarial attacks have received much less attention. The adversarial patch attack was introduced by Brown et al. [13] and focused on physical and universal patches to induce targeted misclassification. Attacks in the real-world can be realized by attaching a patch to the victim object. A follow-up paper on Localized and Visible Adversarial Noise (LaVAN) attack [14] aimed at inducing both targeted and untargeted misclassification in the digital domain. Both of these papers operated in the white-box threat model, with access to the internals of the classifier under attack. PatchAttacker [38], on the other hand, proposed a reinforcement learning-based targeted attack to generate an adversarial patch in a black-box setting. In this paper, we focus on provable defenses against any white-box untargeted attacks such as LaVAN, which we find to be the most effective attack.

Localized patch attacks against object detection [36], [37] and semantic segmentation models [39] as well as training-time poisoning attacks using localized triggers [40], [41] have been proposed. Our threat model in this paper focuses on attacks against classification models at test time, but we expect our defense can be generalized to the above settings as well.

B. Adversarial Patch Defenses

Several defenses have been proposed to mitigate the adversarial patch attack. Digital Watermark (DW) [16] generated a saliency map to detect important pixels and then tried to remove adversarial pixels. However, its detection mechanism is vulnerable to Backward Pass Differentiable Approximation (BPDA) attack [11], which could trick the model into removing benign pixels [18]. Local Gradient Smoothing (LGS) [17] smoothed the suspicious areas of the images to neutralize the effect of the adversarial pixels, but has been bypassed via incorporating the smoothing function into the attack optimization objective [18].

Observing the ineffectiveness of DW and LGS against an adaptive attacker, Chiang et al. [18] proposed the first certified defense against adversarial patch via Interval Bound Propagation (IBP) [19], [20] which traces the influence of adversarial pixels on the hidden representation of each layer and derives a bound for the certified prediction. Despite its important theoretical contribution, the IBP defense has poor clean and provable robust accuracies. For a 5x5 adversarial patch on CIFAR-10 images, IBP only achieves a 47.8% clean accuracy and 30.8% provable robust accuracy, while our Mask-DS has a 80.3% clean accuracy and 61.3% provable robust accuracy. Zhang et al. [21] proposed to use clipped BagNet (CBN) to achieve provable robustness, but their provable accuracy is poor on ImageNet (7.0% for a 2% pixel patch, compared with our 17.7% from Mask-DS). We have shown that CBN is an instance of our general defense framework (Section VI-A), and our robust masking has a better defense performance (Section V). De-randomized Smoothing (DS) [22] proposed building a ‘smoothed’ classifier that outputs the class with the largest count from local predictions on all small pixel patches as the prediction output. DS significantly improved the clean and provable robust accuracy for ImageNet. However, we have also shown in Section VI-A that DS is an instance of our defense framework, we can achieve tighter provable robustness bounds. Moreover, our evaluation in Section V shows that our Mask-BN outperforms DS in terms of clean accuracy and provable robust accuracy. The Minority Report (MR) [26] defense was proposed in concurrent work, where the defender puts a mask at all possible locations and extracts patterns from model predictions on all masked images to detect the adversarial attack. This method incurs a large computational cost by doing inference on all possible masked images and does not scale to ImageNet. Further, this defense only detects the presence of an attack, rather than recovering the correct prediction. This design allows the adversary to force the model to abstain from prediction while our model can always make correct prediction on certified images.
C. Receptive Fields of Convolutional Networks

Several works have studied the influence of the receptive field on model performance in order to better understand the model behavior. BagNet adopted the structure of ResNet-50 but reduced the receptive field size by replacing 3x3 kernels with 1x1 kernels. BagNet-17 can achieve similar validation accuracy as AlexNet on ImageNet dataset when each feature only looks at a 17x17 pixel region. The small receptive field was used for better interpretability of model decisions in the original BagNet paper. In this work, we use the reduced receptive field size for the model robustness to adversarial patch attacks.

D. Other Adversarial Example Attacks and Defenses

Adversarial example attacks and defenses have been an extremely active research area over the past few years. Conventional adversarial attacks craft adversarial examples that have a small $L_p$ distance to normal examples but will induce model misclassification. Many empirical defenses have been proposed to address the adversarial example vulnerability, but most of them can be easily bypassed by a strong adaptive attacker. The fragility of the empirical defenses has inspired provable or certified defenses. The work called PatchGuard that mitigates localized adversarial examples, in Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security (CCS), 2017, pp. 135–147. In this paper, we propose a general provable defense framework called PatchGuard that mitigates localized adversarial patch attacks. We identify large receptive fields and insecure aggregation functions in conventional convolutional networks as the source of vulnerability against adversarial patches. To address these two problems, our defense reduces the size of the receptive field to limit the number of features corrupted by the adversary and further uses a robust masking defense to detect and mask the corrupted features for secure aggregation. Our defense approach achieves state-of-the-art provable robust accuracy on ImageNet and CIFAR-10. We hope that our general defense framework will inspire further research to fully mitigate adversarial patch attacks.

REFERENCES

[1] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. J. Goodfellow, and R. Fergus, “Intriguing properties of neural networks,” in 2nd International Conference on Learning Representations (ICLR), 2014.

[2] I. J. Goodfellow, J. Shlens, and C. Szegedy, “Explaining and harnessing adversarial examples,” in 3rd International Conference on Learning Representations (ICLR), 2015.

[3] N. Papernot, P. D. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, and A. Swami, “The limitations of deep learning in adversarial settings,” in IEEE European Symposium on Security and Privacy (EuroS&P), 2016, pp. 372–387.

[4] N. Carlini and D. A. Wagner, “Towards evaluating the robustness of neural networks,” in 2017 IEEE Symposium on Security and Privacy (S&P), 2017, pp. 39–57.

[5] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, “Towards deep learning models resistant to adversarial attacks,” in 6th International Conference on Learning Representations (ICLR), 2018.

[6] N. Papernot, P. D. McDaniel, X. Wu, S. Jha, and A. Swami, “Distillation as a defense to adversarial perturbations against deep neural networks,” in IEEE Symposium on Security and Privacy (S&P), 2016, pp. 582–597.

[7] W. Xu, D. Evans, and Y. Qi, “Feature squeezing: Detecting adversarial examples in deep neural networks,” in 25th Annual Network and Distributed System Security Symposium (NDSS), 2018.

[8] D. Meng and H. Chen, “Magnet: A two-pronged defense against adversarial examples,” in Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security (CCS), 2017, pp. 135–147.

[9] J. H. Metzen, T. Genevein, W. Fischer, and B. Bischoff, “On detecting adversarial perturbations,” in 5th International Conference on Learning Representations (ICLR), 2017.

[10] N. Carlini and D. A. Wagner, “Adversarial examples are not easily detected: Bypassing ten detection methods,” in Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security (AIStream), 2017, pp. 3–14.

[11] A. Athalye, N. Carlini, and D. A. Wagner, “Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples,” in Proceedings of the 35th International Conference on Machine Learning (ICML), 2018, pp. 274–282.

[12] F. Tramer, N. Carlini, W. Brendel, and A. Madry, “On adaptive attacks to adversarial example defenses,” arXiv preprint arXiv:2002.08347, 2020.

[13] T. B. Brown, D. Mané, A. Roy, M. Abadi, and J. Gilmer, “Adversarial patch,” in Advances in neural information processing systems workshops (NeurIPS Workshops), 2017.

[14] D. Karmon, D. Zoran, and Y. Goldberg, “LaVAN: Localized and visible adversarial noise,” in Proceedings of the 35th International Conference on Machine Learning (ICML), 2018, pp. 2512–2520.

[15] K. Eykholt, I.Evtimov, E. Fernandes, B. Li, A. Rahmati, C. Xiao, A. Prakash, T. Kohno, and D. Song, “Robust physical-world attacks on deep learning visual classification,” in 2018 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018, pp. 1625–1634.

[16] J. Hayes, “On visible adversarial perturbations & digital watermarking,” in 2018 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPR Workshops), 2018, pp. 1597–1604.

[17] M. Naseer, S. Khan, and F. Porikli, “Local gradients smoothing: Defense against localized adversarial attacks,” in IEEE Winter Conference on Applications of Computer Vision (WACV), 2019, pp. 1300–1307.

[18] P.-Y. Chiang, R. Ni, A. Abdelkader, C. Zhu, C. Studor, and T. Goldstein, “Certified defenses for adversarial patches,” in 8th International Conference on Learning Representations (ICLR), 2020.

[19] S. Gowal, K. Dvijotham, R. Stanforth, R. Bunel, C. Qin, J. Uesato, R. Arandjelovic, T. A. Mann, and P. Kohli, “Scalable verified training for provably robust image classification,” in 2019 IEEE/CVF International Conference on Computer Vision (ICCV), 2019, pp. 4841–4850.

[20] M. Mirman, T. Gehr, and M. T. Vechev, “Differentiable abstract interpretation for provably robust neural networks,” in Proceedings of the 35th International Conference on Machine Learning (ICML), 2018, pp. 3575–3583.

[21] Z. Zhang, B. Yuan, M. McCoyd, and D. Wagner, “Clipped bagnet: Defending against sticker attacks with clipped bag-of-features,” in 3rd Deep Learning and Security Workshop (DLS), 2020.

[22] A. Levine and S. Feizi, “(De)randomized smoothing for certifiable defense against patch attacks,” arXiv preprint arXiv:2002.10733, 2020.

[23] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 770–778.

[24] J. Deng, W. Dong, R. Socher, L. Li, K. Li, and F. Li, “ImageNet: A large-scale hierarchical image database,” in 2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), 2009, pp. 248–255.

[25] W. Brendel and M. Bethge, “Approximating CNNs with bag-of-features models works surprisingly well on ImageNet,” in 7th International Conference on Learning Representations (ICLR), 2019.

[26] M. McCoyd, W. Park, S. Chen, N. Shah, R. Roggenkemper, M. Hwang, J. X. Liu, and D. Wagner, “Minority reports defense: Defending against adversarial patches,” arXiv preprint arXiv:2004.13799, 2020.

[27] fast.ai, “ImageNette: A smaller subset of 10 easily classified classes from imagenet,” https://github.com/fastai/imagenette, 2020.
A. Krizhevsky, “Learning multiple layers of features from tiny images,” https://www.cs.toronto.edu/~kriz/learning-features-2009-TK.pdf, 2009.

[27] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” in Advances in Neural Information Processing Systems (NeurIPS), 2012, pp. 1097–1105.

[28] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” in Advances in Neural Information Processing Systems (NeurIPS), 2012, pp. 1097–1105.

[29] S. Thys, W. V. Ranst, and T. Goedemé, “Fooling automated surveillance cameras: Adversarial patches to attack person detection,” in IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPR Workshops), 2019, pp. 49–55.

[30] W. Brendel, “Pretrained bag-of-local-features neural networks,” https://github.com/wielandbrendel/bag-of-local-features-models, 2020.

[31] Y. Liu, S. Ma, Y. Aafer, W. Lee, J. Zhai, W. Wang, and X. Zhang, “Understanding the effective projective fields of neurons in convolutional neural networks?” arXiv preprint arXiv:1909.09469, 2020.

[32] D. Cullina, A. N. Bhagoji, and P. Mittal, “Pac-learning in the presence of adversaries,” in Advances in Neural Information Processing Systems (NeurIPS), 2019, pp. 230–241.

[33] A. Levine and S. Feizi, “Code for the paper “(de)randomized smoothing for certifiable defense against patch attacks,” https://github.com/wielandbrendel/bag-of-local-features-models, 2020.

[34] D. Cullina, A. N. Bhagoji, and P. Mittal, “Pac-learning in the presence of adversaries,” in Advances in Neural Information Processing Systems (NeurIPS), 2019, pp. 230–241.

[35] E. Dohmatob, “Generalized no free lunch theorem for adversarial robustness,” in Proceedings of the 36th International Conference on Machine Learning, 2019, pp. 1646–1654.

[36] D. Yin, R. Kannan, and P. Bartlett, “Rademacher complexity for adversarially robust generalization,” in Proceedings of the 36th International Conference on Machine Learning, 2019, pp. 7085–7094.

[37] L. Schmidt, S. Santurkar, D. Tsipras, K. Talwar, and A. Madry, “Adversarially robust generalization requires more data,” in Advances in Neural Information Processing Systems, 2018, pp. 5014–5026.

[38] N. Papernot, P. McDaniel, A. Sinha, and M. P. Wellman, “Sok: Security and privacy in machine learning,” in 2018 IEEE European Symposium on Security and Privacy (EuroS&P), 2018, pp. 399–414.

[39] X. Yuan, P. He, Q. Zhu, and X. Li, “Adversarial examples: Attacks and defenses for deep learning,” IEEE transactions on neural networks and learning systems, vol. 30, no. 9, pp. 2805–2824, 2019.

[40] I. Feinerer and K. Hornik, wordnet: WordNet Interface, 2017, r package version 0.1-14. [Online]. Available: https://CRAN.R-project.org/package=wordnet

APPENDIX

A. Details of Experiment Setup

ImageNet. ImageNet [24] is a popular benchmark dataset for high-resolution image classification. It has 1281167 training images and 50000 validation images from 1000 classes organized according to the WordNet [57] hierarchy. It is conventional to resize and crop ImageNet images to 224×224 or 299×299. For ResNet, we take 224×224 images as inputs and use the pre-trained model from [34]. For BagNet, we take 224×224 inputs and use pre-trained model from [32]. For DS-ResNet, we uses 299×299 images and use pre-trained models from [33]. The actual input of the base classifier of DS-ResNet is a pixel patch in shape of 25×299; DS-ResNet counts the predictions from all possible 25×299 pixel patches and predicts the class with the largest count.

ImageNet. ImageNet [27] is a subset of ImageNet with 9469 training images and 3925 validation images from classes of tench, English springer, cassette player, chain saw, church, French horn, garbage truck, gas pump, golf ball, parachute. We use this smaller dataset for a more comprehensive defense evaluation. We use the same model architectures as the models for ImageNet, but modify the last fully-connected layer (i.e., the classification layer) to accommodate for 10-class classification. We use the pre-trained models [32]–[34], retrain the entire model for 20 epochs, and retain the model with the highest validation accuracy. We use Stochastic Gradient Descent (SGD) with a 0.001 initial learning rate and a 0.9 momentum for model training. We reduce the learning rate with a factor of 0.1 every 7 epochs.

CIFAR-10. CIFAR-10 [28] is a benchmark dataset for low-resolution image classification. CIFAR-10 has 50000 training images and 10000 test images from classes of airplane, car, bird, cat, deer, dog, frog, horse, ship, truck. Each image is in the shape of 32×32. For ResNet, we train the model from scratch for 20 epochs. We use SGD with a 0.01 initial learning rate, a 0.9 momentum, and a 5e-4 weight decay. We reduce the learning rate with a factor of 0.1 every 10 epochs. For BagNet, we re-scale the 32x32 images to 192x192.
with bicubic interpolation because we find BagNet does not have good performance on low-resolution images. We use the models \cite{32} pre-trained on ImageNet and retrain the entire model with the same hyperparameters as we retrain a BagNet. For DS-ResNet, we take a 4x32 pixel band as the input to its base classifier and uses the pre-trained model from \cite{33}. 