PatchGuard: A Provably Robust Defense against Adversarial Patches via Small Receptive Fields and Masking

Chong Xiang
Princeton University

Arjun Nitin Bhagoji
University of Chicago

Vikash Sehwag
Princeton University

Prateek Mittal
Princeton University

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, and defending against such attacks is an unsolved/open problem. In this paper, we propose a general defense framework called PatchGuard that can achieve high provable robustness while maintaining high clean accuracy against localized adversarial patches. The cornerstone of PatchGuard involves the use of CNNs with small receptive fields to impose a bound on the number of features corrupted by an adversarial patch. Given a bounded number of corrupted features, the problem of designing an adversarial patch defense reduces to that of designing a secure feature aggregation mechanism. Towards this end, we present our *robust masking* defense that robustly detects and masks corrupted features to recover the correct prediction. Notably, we can prove the robustness of our defense against any adversary within our threat model. Our extensive evaluation on ImageNet, ImageNette (a 10-class subset of ImageNet), and CIFAR-10 datasets demonstrates that our defense achieves state-of-the-art performance in terms of both provable robust accuracy and clean accuracy.[1]

1 Introduction

Machine learning models are vulnerable to evasion attacks, where an adversary introduces a small perturbation to a test example for inducing model misclassification [17, 50]. Many prior attacks and defenses focus on the classic setting of adversarial examples that have a small L_p distance to the benign example [2, 7, 8, 17, 33, 35, 36, 41, 42, 50, 52, 56]. However, in the physical world, the classic 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 small restricted area such that the perturbation can be realized by attaching an adversarial patch to the victim object. Several effective patch attacks have been shown: 1) Brown et al. [6] generate physical adversarial patches that can force model predictions to be a target class of the attacker's choice; 2) Karmon et al. [22] propose the LaVAN attack in the digital domain; 3) Eykholt et al. [15] demonstrate a robust physical-world attack that attaches small stickers to a stop sign for fooling traffic sign recognition.

The success of practical localized adversarial patches has inspired several defenses. Digital Watermark (DW) [20] aims to detect and remove the adversarial patch while Local Gradient Smoothing (LGS) [39] proposes smoothing 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 [9]. This has led to the development of several certifiably robust defenses. Chiang et al. [9] propose the first certified defense against adversarial patches via Interval Bound Propagation (IBP) [18, 38]. Zhang et al. [59] use a clipped BagNet (CBN) to achieve provable robustness while Levine et al. [28] propose De-randomized Smoothing (DS) to further improve provable robustness. These works have taken important steps towards provably robust models. However, their performance is still limited in terms of provable robustness and standard classification accuracy (i.e., clean accuracy), leaving defenses against adversarial patches an unsolved/open problem.

1.1 Contributions

In this paper, we propose a general defense framework called PatchGuard that achieves *substantial state-of-the-art provable robustness while maintaining high clean accuracy against localized adversarial patches*.

Insight: Leverage CNNs with Small Receptive Fields. The cornerstone of our defense framework involves the use of Convolutional Neural Networks (CNNs) with small receptive fields to impose a bound on the number of features that can

[1] Our code is available at https://github.com/inspire-group/PatchGuard for the purpose of reproducibility.

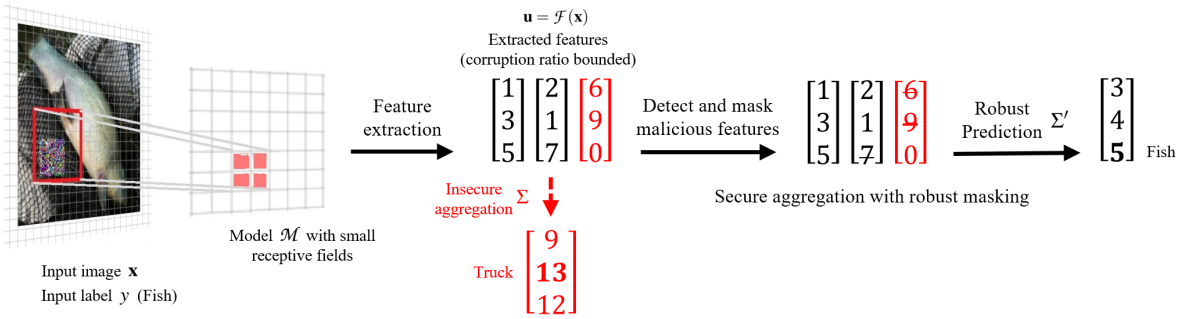

Figure 1: **Overview of defense**. The small receptive field bounds the number of corrupted features (one out of three vectors in this example). The one corrupted feature (red vector) in this example has an abnormally large element that dominates the insecure aggregation (S) but also leads to a distinct pattern from clean features. Our *robust masking* aggregation detects and masks the corrupted feature, recovering the correct prediction from the remaining features. We note that *robust masking* can have false positives (FP) and incorrectly mask benign features, but we show in Section 5 that our defense retains high clean accuracy and provable robust accuracy.

be corrupted due to an adversarial patch. The receptive field of a CNN is the region of an input image that a particular feature is influenced by, and model prediction is based on the aggregation of features extracted from different regions of an image. An example of the receptive field is shown as the red box on the image in Figure 1. Our case study in Section 3.1 demonstrates that a large receptive field makes CNNs more vulnerable to adversarial patch attacks. For a model with a large receptive field of 483 483 (ResNet-50 [21]) on ImageNet images [12], a small patch is present in the receptive field of most extracted features and can thus easily change model prediction. A small receptive field, on the other hand, limits the number of corrupted features, and we use it as the fundamental building block of robust classifiers. We note that a small receptive field is not a barrier to achieving high clean accuracy. A ResNet-like architecture with a small 17 17 receptive field can achieve an AlexNet-level accuracy for ImageNet top-5 classification [5]. The potential robustness improvement, as well as the moderate accuracy drop, motivates the use of small receptive fields in PatchGuard.

Insight: Leveraging Secure Aggregation & Robust Masking. However, a small receptive field alone is not enough for robust prediction since conventional models use insecure feature aggregation mechanisms such as mean. The use of small receptive fields turns the problem of designing an adversarial patch defense into a secure aggregation problem, and we propose *robust masking* as an effective instance of secure feature aggregation mechanism. Figure 1 provides an overview of our defense. The small receptive field ensures that only a small fraction of extracted features are corrupted due to an adversarial patch. The small number of corrupted features forces the adversary to create abnormally large feature values to dominate the final prediction, and *robust masking* aims to detect and mask these abnormal features. Our empirical analysis demonstrates that removing a small number of features of a clean image is unlikely to change model prediction. Therefore, robust masking recovers the correct prediction with high probability if all the corrupted features are masked.

Provable Robustness. Robust masking introduces a fundamental dilemma for the adversary: either to generate conspicuous malicious features that will be detected and masked by our defense or to do with stealthy but ineffective adversarial patches. In Section 4, we show that this dilemma leads to a proof of *provable robustness* for our defense, providing the guarantee that the model can always recover correct predictions on certified images against any adversarial patch within the threat model. This is a stronger notion of robustness compared with defenses that only detect the adversarial attack [34, 35, 56]. We also show that PatchGuard subsumes several existing defenses [28, 59] (as shown in Section 6.1), and outperforms them due to the use of *robust masking*.

State-of-the-art Performance. We consider the strongest adversarial patch attacker, who can place the adversarial patch on any part of the image, including on top of salient objects. We evaluate our provable defense against any patch attacker on ImageNet [12], ImageNette [16], CIFAR-10 [23], and shows that our defense achieves state-of-the-art performance in terms of provable robustness and clean accuracy compared to previous defenses [9, 28, 59]. Our main contributions can be summarized as follows:

1. We demonstrate the use of a small receptive field as a fundamental building block for robustness and leverage it to develop our general defense framework called PatchGuard. PatchGuard is flexible and general as it is compatible with any CNN with small receptive fields and any secure aggregation mechanism.

2. We present *robust masking* as an instance of the secure aggregation mechanism that leads to provable robustness and recovers correct predictions for certified images against any attacker within the threat model.

3. We comprehensively evaluate our defense across ImageNet [12], ImageNette [16], CIFAR-10 [23] datasets, and demonstrate state-of-the-art provable robust accuracy and clean accuracy of our defense.

2 Problem Formulation

In this section, we first introduce the image classification model, followed by the adversarial patch attack and defense formulation. Finally, we present important terminology used in PatchGuard. Table 1 provides a summary of our notation.

2.1 Image Classification Model

We focus on Fully Convolutional Neural Networks (FCNNs) such as ResNet [21], which use convolutional layers for feature extraction and *only one* additional fully-connected layer for the final classification. This structure is widely used in state-of-the-art image classification models [21, 47–49].

We use $\mathcal{X} = [0;1]^{W \times H \times C}$ to denote the image space where each image has width W, height H, number of channels C, and the pixels are re-scaled to $[0;1]$. We take $\mathcal{Y} = \{0;1;\ldots;N-1\}$ as the label space, where the number of classes is N. We use $M(\mathbf{x}) : \mathcal{X} \to \mathcal{Y}$ to denote the model that takes an image $\mathbf{x} \in \mathcal{X}$ as input and predicts the class label $y \in \mathcal{Y}$. We let $F(\mathbf{x}) : \mathcal{X} \to \mathcal{U}$ be the feature extractor that outputs the feature tensor $\mathbf{u} \in \mathcal{U} = \mathbb{R}^{W' \times H' \times C'}$, where W', H', C' are the width, height, and number of channels in this feature map, respectively.

2.2 Attack Formulation

Attack objective. We focus on evasion attacks against an image classification model. Given a deep learning model M, an image $\mathbf{x}$, and its true class label y, the goal of the attacker is to find an image $\mathbf{x}' \in \mathcal{A}(\mathbf{x}) \subset \mathcal{X}$ satisfying a constraint $\mathcal{A}$ such that $M(\mathbf{x}') \neq y$. The constraint $\mathcal{A}$ is defined by the attacker's threat model, which we will describe below. We note that the attack objective of inducing misclassification into any wrong class is referred to as an *untargeted attack*. In contrast, when the goal is to misclassify the image to a particular target class $y' \neq y$, it is called a *targeted attack*. The untargeted attack is easier to launch and thus more difficult to defend against. In this paper, we focus on defenses against the untargeted attack.

Attacker capability. The attacker can arbitrarily modify pixels within a restricted region, and this region can be anywhere on the image, even over the salient object. We assume that all manipulated pixels are within a contiguous region, and the defender has a conservative estimate (i.e., upper bound) of the region size. We note that this matches the strongest threat model used in the existing literature on certified defenses against adversarial patches [9, 28, 59].[2] Formally, we use a binary *pixel block* $\mathbf{p} \in \mathcal{P} = \{0;1\}^{W \times H}$ to represent the restricted region, where the pixels within the region are set to 1. Then, the constraint set $\mathcal{A}(\mathbf{x})$ can be expressed as $\{\mathbf{x}' = (\mathbf{1} - \mathbf{p}) \odot \mathbf{x} + \mathbf{p} \odot \mathbf{x}'' \mid \mathbf{x}, \mathbf{x}' \in \mathcal{X}; \mathbf{x}'' \in [0;1]^{W \times H \times C}; \mathbf{p} \in \mathcal{P}\}$, where $\odot$ refers to the element-wise product operator, and $\mathbf{x}''$ is the content of the adversarial patch. In this paper, we primarily focus on the case where $\mathbf{p}$ represents one square region. Our defense can generalize to other shapes and we defer experimental results for this to our technical report [55].

[2] A high-performance provable defense against a single patch is currently an open/unsolved problem and is thus the focus of our threat model. We will discuss our defense extension for multiple patches in Appendix E.

Table 1: Table of notation

Notation	Description
$\mathcal{X} = [0;1]^{W \times H \times C}$	Image space
$\mathcal{Y} = \{0;1;\ldots;N-1\}$	Label space
$\mathcal{U} = \mathbb{R}^{W' \times H' \times C'}$	Feature space
$M(\mathbf{x}) : \mathcal{X} \to \mathcal{Y}$	Model predictor from $\mathbf{x} \in \mathcal{X}$
$F(\mathbf{x}) : \mathcal{X} \to \mathcal{U}$	Local feature extractor for all classes
$F(\mathbf{x}; l) : \mathcal{X} \to \mathcal{U}$	Local feature extractor for class l
$\mathcal{P} = \{0;1\}^{W \times H}$	Set of binary pixel blocks in the image space
$\mathcal{W} = \{0;1\}^{W' \times H'}$	Set of binary windows in the feature space

2.3 Defense Formulation

Defense objective. The goal of our defense is to design a defended model D such that $D(\mathbf{x}) = D(\mathbf{x}') = y$ for any clean data point $(\mathbf{x}; y) \in \mathcal{X} \times \mathcal{Y}$ and any adversarial example $\mathbf{x}' \in \mathcal{A}(\mathbf{x})$, where $\mathcal{A}(\mathbf{x})$ is the adversarial constraint introduced in Section 2.2. Note that we aim to *recover the correct prediction*, which is harder than merely detecting an attack.

Provable robustness. Previous works [7, 9, 52] have shown that empirical defenses are usually vulnerable to an adaptive white-box attacker who has full knowledge of the defense algorithm, model architecture, and model weights; therefore, we design PatchGuard as a provably robust defense [9, 10, 18, 28, 38, 59] to provide the strongest robustness. *The evaluation of provable defense is agnostic to attack algorithms and its result holds for any attack considered in the threat model.*

2.4 PatchGuard Terminology

Local feature and its receptive field. Recall that we use F to extract feature map as $\mathbf{u} \in \mathbb{R}^{W' \times H' \times C'}$. We refer to each $1 \times 1 \times C'$-dimensional feature in tensor $\mathbf{u}$ as a *local feature* since it is only extracted from part of the input image as opposed to the entire image. We define the *receptive field* of a local feature to be a subset of image pixels that the feature $\tilde{\mathbf{u}} \in \mathbb{R}^{1 \times 1 \times C'}$ is looking at, or affected by. Formally, if we represent the input image x as *a set of pixels*, the receptive field of a particular local feature $\tilde{\mathbf{u}}$ is a subset of pixels for which the gradient of $\tilde{\mathbf{u}}$ is non-zero, i.e., $\{r \in x \mid \tilde{\nabla}_r \tilde{\mathbf{u}} \neq \mathbf{0}\}$. For simplicity, we use the phrase "receptive field of a CNN" to refer to "receptive field of a particular feature of a CNN".

Global feature and global logits. When the local feature tensor $\mathbf{u}$ is the output of the last convolutional layer, conventional CNNs use an element-wise linear aggregation (e.g., mean) over all local features to obtain the *global feature* in $\mathbb{R}^{C'}$. The global feature will then go through the last fully-connected layer (i.e., classification layer) and yield the *global logits* vector in $\mathbb{R}^N$ for the final prediction (top of Figure 2).

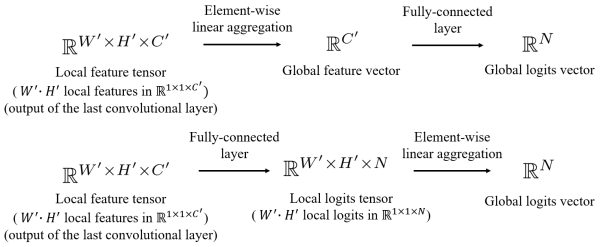

Figure 2: Two equivalent ways of computing the global logits vector (top: used in conventional CNNs; bottom: used in our defense).

Local logits. Similar to computing the global logits from the global feature, we can feed each local feature (in $\mathsf{R}^{1\ 1\ C'}$) to the fully-connected layer to get the *local logits* (in $\mathsf{R}^{1\ 1\ N}$). Each local logits vector is the classification output based on each local feature; thus, they share the same receptive field. Concatenating all $W' \ H'$ local logits vectors gives the local logits tensor, and applying the element-wise linear aggregation gives the same global logits (bottom of Figure 2).

Local confidence, local prediction, and class evidence. Based on local logits, we can derive the concept of *local confidence* and *local prediction* tensor by feeding the local logits tensor to a softmax layer and an argmax layer, respectively. In the remainder of this paper, we *specialize the concept of feature* by considering it to refer to either a logits tensor, a confidence tensor, or a prediction tensor. In this case, we have $C' = N$. We also sometimes abuse the notation by letting $F(\mathbf{x}; l) : X \ Y \ ! \ \mathsf{R}^{W' \ H'}$ denote the slice of the feature corresponding to class l. We call the elements of $F(\mathbf{x}; l)$ the *class evidence* for class l.

3 PatchGuard

In this section, we first use an empirical case study to motivate the use of small receptive fields and secure feature aggregation (i.e., *robust masking*). Next, we will give an overview of our general PatchGuard framework, followed by our use of networks with small receptive fields and details of our *robust masking* based secure aggregation. The provable robustness of this defense will be demonstrated and analyzed in Section 4.

3.1 Why are adversarial patches effective?

Previous work [6, 22] on adversarial patches, surprisingly, shows that model prediction can be manipulated by patches that occupy a very small portion of input images. In this subsection, we provide a case study for ResNet-50 [21] trained on ImageNet [12], ImageNette (a 10-class subset of ImageNet) [16], and CIFAR-10 [23] datasets and identify two critical reasons for the model vulnerability. These will then motivate the development and discussion of our defense.

Experiment setup. We take 5000 random ImageNet validation images and the entire validation sets of ImageNette and CIFAR-10 for the case study. We use a patch consisting of 3% of the image pixels for an empirical attack. Further details about the attack setup and datasets are covered in our technical report [55]. We extract the local logits (as defined in Section 2.4) from adversarial images for further analysis.

Table 2: Percentage of incorrect predictions of ResNet-50

Dataset Patch size	ImageNet 3% pixels	ImageNette 3% pixels	CIFAR-10 3% pixels
Incorrect local pred. (attacked)	84.4%	56.4%	67.0%
Incorrect local pred. (original)	59.9%	15.3%	27.0%
Incorrect local pred. (difference)	24.5%	41.1%	40.0%
Incorrect global predictions	99.9%	99.1%	95.5%

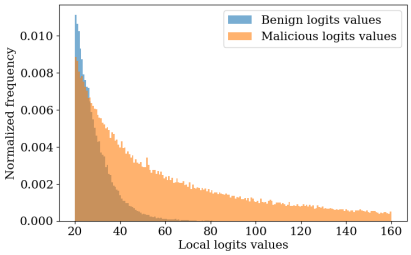

Figure 3: Histogram of large local logits values for ImageNet adversarial images (only positive values larger than 20 are shown).

Vulnerability I: the small adversarial patch appears in the large receptive fields of most local features and is able to manipulate the local predictions. In Table 2, we report the percentage of incorrect local predictions of the adversarial images (attacked) and clean images (original) as well as their percentage difference. We can see that a small patch that only takes up 3% of the image pixels can corrupt 24.5% additional local predictions for ImageNet images, 41.1% for ImageNette, and 40.0% for CIFAR-10. As shown in the table, the large portion of incorrect local predictions finally leads to a high percentage of incorrect global predictions. This vulnerability mainly stems from the large receptive field of ResNet-50. Each local feature of ResNet-50 is influenced by a 483 483 pixel region in the input space (with zero padding) [1]; therefore, even if the adversarial patch only appears in a small restricted area, it is still within the receptive field of many local features and can manipulate the local predictions.[3] This observation motivates the use of small receptive fields: if the receptive field is small, it ensures that only a limited number of local features can be corrupted by an adversarial patch, and robust prediction may be possible.

Vulnerability II: the adversarial patch creates large malicious local feature values and makes linear feature aggregation insecure. In Figure 3, we plot the histogram of

[3] We note that a patch appearing in the receptive field of a local feature does not necessarily indicate a successful local feature corruption. Each local feature focuses exponentially more on the center of its receptive field (further details are in Appendix B). When the adversarial patch is far away from the center of the receptive field, its influence on the feature is greatly limited.

class evidence of the true class and the malicious class of the adversarial images from ImageNet (we report similar results for other two datasets in our technical report [55]). 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 CNNs use simple linear operations such as average pooling to aggregate all local features, and thus are vulnerable to these large malicious feature values. This observation motivates our development of robust masking as a secure feature aggregation mechanism.

3.2 Overview of PatchGuard

In Section 3.1, we identified the large receptive field and insecure aggregation of conventional CNNs as two major sources of model vulnerability. In this subsection, we provide an overview of our defense that tackles both problems.

Recall that Figure 1 provides an overview of our defense framework. We consider a CNN M with small receptive fields. The feature extractor $F(x)$ produces the local feature tensor u extracted from the input image x, where u can be any one of the logits, confidence, or model prediction tensor. Our defense framework is compatible with any CNN with small receptive fields, and we will present two general ways of building such networks in Section 3.3. 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 summation might still result in a misclassification. To address this vulnerability, we propose a robust masking algorithm for secure feature aggregation.

In robust masking, we detect and mask the corrupted features in the local feature tensor $u = F(x)$. Since the number of corrupted local features is limited due to the small receptive field, the adversary is forced to create large feature values to dominate the global prediction. These large feature values lead to a distinct pattern and enable our detection of corrupted features. Further, we empirically find that that model predictions are generally invariant to the removal of partial features (Section 5.3.1). Therefore, once the corrupted features are masked, we are likely to recover the correct prediction y with the remaining local features (right part of Figure 1). This defense introduces a dilemma for the adversary: either to generate conspicuous malicious features that will be detected and masked by our defense or to use stealthy but ineffective adversarial patches. This fundamental dilemma enables provable robustness. We will introduce the details of robust masking in Section 3.4, and perform its provable analysis in Section 4.

3.3 CNNs with Small Receptive Fields

Our defense framework is compatible with any CNN with small receptive fields[4]. In this subsection, we discuss two general ways to build such CNNs; our goal is to reduce the number of image pixels that can affect a particular feature.

Building an ensemble model. One approach to design a network with small receptive fields is to divide the original image into multiple small pixel patches and feed each pixel patch to a base model for separate classification. We can then build an ensemble model aggregating the output of base models. In this ensemble model, a local feature is the base model output, which can be logits, confidence, or prediction. Since the base model only takes a small pixel patch as input, each local feature is only affected by a small number of pixels, and thus the ensemble model has a small receptive field. We note that as the image resolution becomes higher, the number of all possible pixel patches increases greatly, which leads to a huge training and testing computation cost of the ensemble model. A natural approach to reduce the computation cost is to do inference on a sub-sampled set of small pixel patches.

Using small convolution kernels. A more efficient approach is to use small convolution kernels in conventional CNN architectures. In Figure 4, we provide an illustration for 1-D convolution computation with different kernel sizes. As we can see, the output cell is affected by all 5 input cells when using two convolutions with a kernel size of 3 (left) while each output cell is only affected by 3 input cells when reducing the size of one kernel to 1 (right). This logic extends directly to the large CNNs used in practice by replacing large convolution kernels with small kernels. Moreover, we can use a convolution stride to skip a portion of small pixel patches to reduce the computation cost. The modified CNN can be regarded as an ensemble model from a subset of all possible pixel patches. With this formulation, we can efficiently extract all local features with one-time model feed-forward computation. In Section 5, we will instantiate both approaches by adapting the implementation from Levine et al. [27] and Brendel et al. [4] and compare their performance.

Remark: translation from images into features. The use of CNNs with small receptive fields translates the adversarial patch defense problem from the image space to the feature space. That is, the problem becomes one of performing robust prediction from the feature space where a limited-size contiguous region is corrupted (due to a limited-size contiguous adversarial patch in the image space). The security analysis in the feature space (i.e., local logits, confidence, or prediction tensor) is simplified due to the use of linear aggregation, in

Figure 4: Effect of the convolution kernel size on the output receptive field size (left: two convolutions with a kernel size of 3; right: two convolutions with a kernel size of 1 and 3, respectively).

[4] The receptive field should be small compared with the input image size.

contrast with the high non-linearity of CNN models if we directly analyze the input image. This observation enables our robust masking technique as well as our provable analysis.

3.4 Robust Masking

Given that an adversarial patch can only corrupt a limited number of local features with small receptive fields, the adversary is forced to create a small region of abnormally high feature values to induce misclassification. In order to detect this corrupted region, we clip the feature values and use a sliding window to find the region with the highest class evidence for each of the classes. We then apply a mask to the suspected region for each class so that the final classification is not influenced by the adversarial features. The defense algorithm is shown in Algorithm 1.

Algorithm 1 Robust masking

Input: Image x, label space $\mathcal{Y}$, feature extractor F of model M, clipping bound $[c_l; c_h]$, the set of sliding windows $\mathcal{W}$, and detection threshold $T \in [0; 1]$. Default setting: $c_l = 0; c_h = \infty; T = 0$.
Output: Robust prediction y

```
procedure RobustMasking
    for each ȳ ∈ Y do
        u_ȳ ← F(x, ȳ)                    . Local feature for class ȳ
        û_ȳ ← Clip(u_ȳ, c_l, c_h)        . Clipped local features
        w_ȳ ← Detect(û_ȳ, T, W)          . Detected window
        s_ȳ ← Sum(û_ȳ ⊙ (1 − w_ȳ))       . Applying the mask
    end for
    y ← arg max_{ȳ∈Y} (s_ȳ)
    return y
end procedure

procedure Detect(û_ȳ, T, W)
    w_ȳ ← arg max_{w∈W} Sum(w ⊙ û_ȳ)     . Detection
    b ← Sum(w_ȳ ⊙ û_ȳ) / Sum(û_ȳ)        . Normalization
    if b ≤ T then
        w_ȳ ← 0                          . An empty mask returned
    end if
    return w_ȳ
end procedure
```

Clipping. As shown in Algorithm 1, our defense will iterate over all possible classes in $\mathcal{Y}$. For each class $\bar{y}$, we first get its corresponding clipped local feature tensor $\hat{u}_{\bar{y}}$ from the undefended model. We set the default values of the clipping bounds to $c_l = 0; c_h = \infty$ for all feature types and datasets. When the feature type is logits, we clip the negative values to zero since our empirical analysis in Section 5.3.1 shows that they contribute little 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 is a confidence tensor or one-hot encoded prediction, it is unaffected by clipping, since its values are already bounded in $[0; 1]$.

Feature windows. We use a sliding window to detect and mask the abnormal region in the feature space. A window is a binary mask in the feature space whose size matches the upper bound of the number of local features that can be corrupted by the adversarial patch. Formally, let p be the upper bound of patch size in the threat model, r be the size of receptive field, and s be the stride of receptive field, which is the pixel distance between two adjacent receptive centers. We can compute the optimal window size as

$$w = \lceil (p + r - 1)/s \rceil \tag{1}$$

This equation can be derived by considering the worst-case patch location and counting the maximum number of corrupted local features. A detailed derivation is in Appendix B.

We note that the window size is a tunable security parameter and we use a conservative window size (computed with the upper bound of the patch size) to make robust masking agnostic to the actual patch size used in an attack. The implications of using an overly conservative window size are discussed in Section 5.3.2 and Appendix C. We represent each window w with a binary feature map in $\{0; 1\}^{W' \times H'}$, where features within the window have values of one.

Detection. We use the subprocedure Detect to examine the clipped local feature tensor $\hat{u}_{\bar{y}}$ and detect the suspicious region. Detect takes the feature tensor $\hat{u}_{\bar{y}}$, the normalized detection threshold $T \in [0; 1]$, and a set of sliding windows $\mathcal{W}$ as inputs. To detect the malicious region, Detect calculates the sum of feature values (i.e., the class evidence) for class $\bar{y}$ within every possible window and identifies the window with the highest sum of class evidence. If the normalized highest class evidence exceeds the threshold T, we return the corresponding window $w_{\bar{y}}$ as the suspicious window for that class; otherwise, we return an empty window 0.

Masking. If we detect a suspicious window in the local feature space, we mask the features within the suspicious area and calculate the sum of class evidence from the remaining features as $s_{\bar{y}} = \text{Sum}(\hat{u}_{\bar{y}} \odot (1 - w_{\bar{y}}))$. After we calculate the masked class evidence $s_{\bar{y}}$ for all possible classes in $\mathcal{Y}$, the defense outputs the prediction as the class with largest class evidence, i.e., $y = \arg\max_{\bar{y} \in \mathcal{Y}} (s_{\bar{y}})$.

4 Provable Robustness Analysis

In this section, we provide provable robustness analysis for our robust masking defense. For any clean image x and a given model M, we will determine whether any attacker, with the knowledge of our defense, can bypass the robust masking defense. Recall that our threat model allows the adversarial patches to be within one restricted region. Given this threat model, all the corrupted features will also be within a small window in the feature map space when using a CNN with small receptive fields; we call this window *malicious window*.

Provable Robustness via an adversary dilemma. With the robust masking defense, we put the adversary in a dilemma. If the adversary wants to succeed in the attack, they need to increase the class evidence of a wrong class. However, increasing the class evidence will trigger our detection and masking mechanism that reduces the class evidence. As a result, this dilemma imposes an upper bound on the class evidence of any class $s_{\bar{y}}$ (in Line 6 of Algorithm 1), which further enables provable robustness. In fact, we can first prove the following lemma.

Lemma 1. Given a malicious window $w \in \mathcal{W}$, a class $\bar{y} \in \mathcal{Y}$, the set of sliding windows $\mathcal{W}$, the clipped and masked class evidence of class $\bar{y}$ (i.e., $s_{\bar{y}}$ in Algorithm 1) can be no larger than $\text{SUM}(\hat{u}_{\bar{y}} \odot (1 - w))/(1 - T)$ when setting $c_l = 0$ and $T \in [0, 1)$.

Proof. The goal of the adversary is to modify the content within the malicious window w to bypass our defense. Let e be the amount of class evidence within w and $t = \text{SUM}(\hat{u}_{\bar{y}} \odot (1 - w))$ be the class evidence outside w. Note that the adversary has control over the value of e but not t, and that the total class evidence of the modified malicious feature tensor is no more than $t + e$. Next, the subprocedure DETECT will take the malicious feature tensor as input and detect a suspicious window $w_{\bar{y}}$. Finally, a mask is applied and the class evidence is reduced to $s_{\bar{y}} = t + e - e'$, where e' is the class evidence within the detected window $w_{\bar{y}}$. To obtain the upper bound of $s_{\bar{y}}$ given a specific malicious window w, we will determine the ranges of e, e' in four possible cases of the detected window $w_{\bar{y}}$ as illustrated in Figure 5.

1. Case I: the malicious window is perfectly detected. In this case, we have $w = w_{\bar{y}}$ and thus $e = e'$. The class evidence $s_{\bar{y}} = t + e - e' = t$.

2. Case II: a benign window is incorrectly detected. In this case, we have $e' = \text{SUM}(\hat{u}_{\bar{y}} \odot w_{\bar{y}})$. The adversary has the constraint that $e \geq e'$, otherwise, the malicious window w instead of $w_{\bar{y}}$ will be detected. Therefore, we have $s_{\bar{y}} = t + e - e' \geq t$.

3. Case III: the malicious window is partially detected. Let $r_1 = w_{\bar{y}} \odot (1 - w)$ be the detected benign region, $r_2 = w_{\bar{y}} \odot w$ be the detected malicious region, and $r_3 = (1 - w_{\bar{y}}) \odot w$ be the undetected malicious region. Let q_1, q_2, q_3 be the class evidence within region r_1, r_2, r_3, respectively. We have $e = q_2 + q_3$ and $e' = q_1 + q_2$. Similar to Case II, the adversary has the constraint that $e \geq e'$, or $q_3 \geq q_1$; otherwise, w instead of $w_{\bar{y}}$ will be detected. Therefore, we have $s_{\bar{y}} = t + e - e' = t + q_3 - q_1 \geq t$.

4. Case IV: no suspicious window detected. This case happens when the largest sum within every possible window does not exceed the detection threshold. We have $e = (e + t) \cdot T$, which yields $e \leq tT/(1 - T)$. We also have $e' = 0$ since no mask is applied. Therefore, the class evidence satisfies $s_{\bar{y}} = t + e - t/(1 - T)$, where $T \in [0, 1]$.

Figure 5: Illustrations for four cases of detected window $w_{\bar{y}}$. The clipped and masked class evidence satisfies $s_{\bar{y}} \leq t + e - e'$. For Case I, II, III, we have $e \geq e'$ and therefore $s_{\bar{y}} \geq t$. For Case IV, we have $e \leq tT/(1 - T)$; $e' = 0$ and therefore $s_{\bar{y}} \geq t/(1 - T)$.

Combining the above four cases, we have the upper bound of the target class evidence to be $t/(1 - T) = \text{SUM}(\hat{u}_{\bar{y}} \odot (1 - w))/(1 - T)$. □

Provable analysis. Lemma 1 shows that robust masking limits the adversary's ability to increase the malicious class evidence. If the upper bound of malicious class evidence is not large enough to dominate the lower bound of the true class evidence, we can certify the robustness of our defense on a given clean image. The pseudocode of our provable analysis is provided in Algorithm 2. Next, we will explain our analysis by proving the following theorem.

Theorem 1. Let $c_l = 0, T \in [0, 1), w \in \mathcal{W}$ denote the sliding windows whose sizes are determined by Equation 1, $A(x)$ denote the adversary's constraint as defined in Section 2.2. If Algorithm 2 returns True for a given image x, our defense in Algorithm 1 can always make a correct prediction on any adversarial image $x' \in A(x)$.

Proof. Our provable analysis in Algorithm 2 iterates over all possible windows $w \in \mathcal{W}$ and all possible target classes $y' \in \mathcal{Y}' = \mathcal{Y} \setminus \{y\}$ to derive provable robustness for the untargeted attack with a patch at any location. For each possible malicious window w, Algorithm 2 determines the upper bound of the class evidence of each target class (Line 3-6) and the lower bound of the class evidence of the true class (Line 7-9).

For each target class y', we can apply Lemma 1 and get the upper bound $s_{y'} = \text{SUM}(\hat{u}_{y'} \odot (1 - w))/(1 - T)$.

For the true class y, the optimal attacking strategy is to set all true class evidence within the malicious window to $c_l = 0$. Note that the true class evidence within the detected window w_y (if any) will be masked. Therefore, the lower bound s_y is equivalent to removing class evidence within w and w_y, i.e., $s_y = \text{SUM}(\hat{u}_y \odot (1 - w) \odot (1 - w_y))$.

The final step is to compare the upper bound of target class evidences $s_{y'}$ with the lower bound of true class evidence s_y

Algorithm 2 Provable analysis of robust masking

Input: Image x, true class y, wrong label set $Y' = Y \setminus \{y\}$, feature extractor F of model M, clipping upper bound c_h, the set of sliding windows W, detection threshold T.
Output: Whether the image x has provable robustness

```
procedure ProvableAnalysisMasking
    for each w ∈ W do
        . Upper bound of target class evidence
        for each y' ∈ Y' do
            û_y' ← Clip(F(x, y'), 0, c_h)
            s̄_y' ← Sum(û_y' − (1 − w))/(1 − T)
        end for
        . Lower bound of true class evidence
        û_y ← Clip(F(x, y), 0, c_h)
        w_y ← Detect(û_y − (1 − w), W, T)
        s_y ← Sum(û_y − (1 − w) − (1 − w_y))
        . Feasibility of an attack
        if max_{y'∈Y'}(s̄_y') > s_y then
            return False
        end if
    end for
    return True
end procedure
```

If the condition $\max_{y' \in Y'}(\bar{s}_{y'}) > s_y$ is satisfied, we assume an attack is possible and the algorithm returns *False*. On the other hand, if Algorithm 2 checks all possible malicious windows $w \in W$ for all possible target classes $y' \in Y'$ and does not return *False* in any case, this means our defense on this clean image has provable robustness against any possible patch and can always make a correct prediction. □

Provable adversarial training. We note that our provable analysis can be incorporated into the training process to improve provable robustness. We call this "provable adversarial training" and will discuss its details in Appendix A.

5 Evaluation

In this section, we provide a comprehensive evaluation of PatchGuard. We report the provable robust accuracy of our defense (obtained from Algorithm 2 and Theorem 1) on the ImageNet [12], ImageNette [16], and CIFAR-10 [23] datasets for various patch sizes. We instantiate our defense with multiple different CNNs with small receptive fields and compare their performance with previous provably robust defenses [9, 28, 59]. We also provide a detailed analysis of our defense performance with different settings.

5.1 Experiment Setup

Datasets. We report our main provable robustness results on the 1000-class ImageNet [12], 10-class ImageNette [16], and 10-class CIFAR-10 [23] datasets. ImageNet and ImageNette images have a high resolution and were resized and cropped to 224×224 or 299×299 before being fed into different models while CIFAR-10 images have a lower resolution of 32×32. CIFAR-10 images are rescaled to 192×192 before being fed to BagNet. Further details are in our technical report [55].

Models. As discussed in Section 3.3, we have two general ways to build a network with small receptive fields. In our evaluation, we instantiate the ensemble approach using a de-randomized smoothed ResNet (DS-ResNet) [28], and the small convolution kernel approach using BagNet [5]. The DS-ResNet [28] takes a rectangle pixel patch, or a pixel band, as the input of its base model and uses prediction majority voting for the ensemble prediction. In contrast, our defense uses robust masking for aggregation. The BagNet [5] architecture replaces a fraction of 3×3 convolution kernels of ResNet-50 with 1×1 kernels to reduce the receptive field size. It was originally proposed in the context of interpretable machine learning while we use this model for provable robustness against adversarial patch attacks.
We analyze performance of ResNet-50, BagNet-33, BagNet-17, BagNet-9, and DS-25-ResNet-50. These 5 models have a similar network structure but have different receptive fields of 483×483, 33×33, 17×17, 9×9, and 25×299, respectively. For CIFAR-10, we additionally include a DS-ResNet-18 with a band size of 4 (DS-4-ResNet-18). Model training details are in our technical report [55].

Defenses. We report the defense performance of our robust masking defense with the BagNet (Mask-BN) and with the DS-ResNet (Mask-DS). We also compare with the existing Clipped BagNet (CBN) [59], De-randomized Smoothing (DS) [28] and Interval Bound Propagation based certified defense (IBP) [9]. The default settings of our defense are listed in Table 3. Note that for PatchGuard, we use the same set of parameters (i.e. c_l, c_h, T) for all datasets and models. For previous defenses, we use the optimal parameter settings obtained from their respective papers.

Attack Patch Size. For ImageNet and ImageNette, we analyze our defense performance against a single square adversarial patch that consists of up to 1%, 2%, or 3% pixels of the images. For CIFAR-10, we report results for a patch consisting of 0.4% or 2.4% of the image pixels. In Appendix F, we analyze the defense performance against larger patches to understand the limits of PatchGuard.

Table 3: Default defense settings for Mask-BN and Mask-DS

Setting	Feature	Parameters
Mask-BN on ImageNet(te)	BagNet-17 logits	$c_l = 0$, $c_h = \infty$, $T = 0$
Mask-BN on CIFAR-10	BagNet-17 logits	
Mask-DS on ImageNet(te)	DS-25-ResNet-50 confidence	
Mask-DS on CIFAR-10	DS-4-ResNet-18 confidence	

Table 4: 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	robust	clean	robust	clean	robust	clean	robust	clean	robust	clean	robust	clean	robust	clean	robust
Mask-BN	95.2	89.0	95.0	86.7	94.8	83.0	55.1	32.3	54.6	26.0	54.1	19.7	84.5	63.8	83.9	47.3
Mask-DS	92.3	83.1	92.1	79.9	92.1	76.8	44.1	19.7	43.6	15.7	43.0	12.5	84.7	69.2	84.6	57.7
IBP [9]	computationally infeasible												65.8	51.9	47.8	30.8
CBN [59]	94.9	74.6	94.9	60.9	94.9	45.9	49.5	13.4	49.5	7.1	49.5	3.1	84.2	44.2	84.2	9.3
DS [28]	92.1	82.3	92.1	79.1	92.1	75.7	44.4	17.7	44.4	14.0	44.4	11.2	83.9	68.9	83.9	56.2

5.2 Provable Robustness Results

In this subsection, we present provable robustness results for our defense (computed with Algorithm 2 and Theorem 1); the results hold for any attack within the corresponding patch size constrain. We also compare PatchGuard with previous provably robust defenses [9, 28, 59].

PatchGuard achieves high provable robustness across different models and datasets. We report the provable robust accuracy of PatchGuard across different models, patch sizes, and datasets in Table 4. First, both Mask-BN and Mask-DS achieve high provable robustness. For example, against a 1% pixel patch on the 10-class ImageNette dataset, Mask-BN has a provable robust accuracy of 89.0% while Mask-DS has that of 83.1%. This implies that for 89.0% and 83.1% of the images from the respective test sets, no attack using a 1% pixel patch can succeed. Second, PatchGuard has high provable robustness across different datasets. Even for the extremely challenging 1000-class ImageNet dataset, Mask-BN achieves a non-trivial provable robust accuracy of 32.3% for the 1% pixel patch. The provable robust accuracy increases to 54.8% if we consider the top-5 classification task (more details for the top-k analysis are in Appendix D).

PatchGuard also maintains high clean accuracy. As shown in Table 4, PatchGuard retains high clean accuracy. For a 1% pixel patch, Mask-BN has a 95.2% clean accuracy on ImageNette and 55.1% on ImageNet. Mask-DS also has a 92.3% clean accuracy on ImageNette and 44.1% on ImageNet. For a 2.4% pixel patch on CIFAR-10, Mask-BN and Mask-DS have a high clean accuracy of 83.9% and 84.6%, respectively. In Table 5, we report the clean accuracy of ResNet and BagNet. We can see that the clean accuracy drop of Mask-BN and Mask-DS on ImageNette compared with undefended ResNet is within 7.5%. The accuracy drop of Mask-BN from the undefended BagNet is within 1%[5].

We note that we use the optimal mask window sizes for different estimated upper bounds of patch sizes, and therefore the clean accuracy for different patches varies slightly in Table 4. We will show a similarly high performance of our defense when using an over-conservatively large mask window size in Section 5.3.2.

[5] BagNet alone does not have any provable robustness but acts as a building block for the provable defense of PatchGuard.

Table 5: Clean accuracy of ResNet and BagNet for different datasets

Dataset	ImageNette	ImageNet	CIFAR-10
ResNet	99.6%	76.1%	97.0%
BagNet	95.9%	56.5%	85.4%

PatchGuard achieves higher provable robust accuracy than all previous defenses. We compare our defense performance with existing defenses across three datasets.

Comparison with IBP [9]. IBP is too computationally expensive and does not scale to high-resolution images like ImageNette and ImageNet. We thus only compare its performance with PatchGuard on CIFAR-10. As shown in Table 4, both Mask-BN and Mask-DS significantly outperform IBP in terms of provable robust accuracy and clean accuracy.

Comparison with CBN [59]. Table 4 shows that both Mask-BN and Mask-DS have higher provable robust accuracy than CBN across three datasets. The clean accuracy of Mask-BN is higher or comparable with that of CBN, but its provable robust accuracy is much higher. For example, against a 3% pixel patch on ImageNette, Mask-BN (94.8%) has a similar clean accuracy as CBN (94.9%), but its provable robust accuracy is 37.1% higher!

Comparison with DS [28]. Both Mask-BN and Mask-DS have better defense performance than DS on the high-resolution ImageNette and ImageNet datasets. For example, against a 1% pixel patch on ImageNet, Mask-BN has a 10.7% higher clean accuracy and a 14.6% higher provable robust accuracy compared with DS. On CIFAR-10, Mask-DS outperforms DS in terms of clean accuracy and provable robust accuracy thanks to the robust masking defense.

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 the 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 robustness, the choice of which model to use (Mask-BN or Mask-DS) depends on the dataset.

Table 6: Effect of logits clipping values on vanilla models

$(c_l; c_h)$	$(-\infty; \infty)$	$(0; \infty)$	$(0; 50)$	$(0; 15)$	$(0; 5)$
ResNet-50	99.6%	99.5%	99.5%	99.5%	99.0%
BagNet-33	97.2%	97.1%	97.0%	95.8%	94.1%
BagNet-17	95.9%	95.5%	94.7%	92.3%	87.9%
BagNet-9	92.5%	92.5%	91.4%	85.4%	73.8%

Table 7: Invariance of BagNet-17 predictions to feature masking

Window size	0 0	2 2	4 4	6 6	8 8
Masked accuracy	95.9%	95.9%	95.9%	95.8%	95.7%
% images	4.1%	5.1%	6.1%	7.3%	8.5%
% windows per image	0%	0.05%	0.2%	0.4%	0.7%

Table 8: Effect of receptive field sizes on provable robust accuracy

Patch size	1% pixels		2% pixels		3% pixels	
Accuracy	clean	robust	clean	robust	clean	robust
Mask-BN-33	96.5%	88.9%	96.3%	86.0%	96.3%	82.1%
Mask-BN-17	95.2%	89.0%	95.0%	86.7%	94.8%	83.0%
Mask-BN-9	92.1%	85.5%	91.8%	82.8%	91.5%	79.8%

Table 9: Effect of detection thresholds on Mask-BN-17

	Clean accuracy	Provable accuracy	Detection FP
T-0.0	95.0%	86.7%	100%
T-0.2	94.2%	79.9%	22.9%
T-0.4	95.3%	68.0%	0.7%
T-0.6	95.5%	38.7%	0.05%
T-0.8	95.5%	6.2%	0%
T-1.0	95.5%	0%	0%

5.3 Detailed Analysis of PatchGuard

In this subsection, we analyze the behavior of vanilla (undefended) models, PatchGuard with different parameters, and defense efficiency on the ImageNette dataset. We will only report results for Mask-BN when the observations from Mask-BN and Mask-DS are very similar. A similar analysis for CIFAR-10 is available in our technical report [55].

5.3.1 Analysis of Vanilla models

Recall that PatchGuard's robust prediction relies on clipping feature values as well as robust masking. Here, we show that vanilla models only have a small performance loss due to clipping and feature masking, which explains the high clean accuracy retained by PatchGuard.

Clipping has a small impact on vanilla models. In this analysis, 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 6. We find that clipping the negative values only slightly affects the clean accuracy ($c_l = 0; c_h = \infty$ is our default setting). 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 logits 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 logits value, we will not affect the clean accuracy and can improve the empirical robustness against the adversarial patch.

Vanilla models are generally prediction-invariant 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 the remaining features, we can recover the correct prediction. We use BagNet-17, which has 26×26 local features, to analyze the prediction invariance of vanilla models to partial feature masking. We mask out all class evidence within a set of sliding windows of different sizes and record the prediction from the remaining features. We report the average accuracy over all possible masked feature tensors (masked accuracy), the percentage of images for which at least one masked prediction is incorrect (% images), and the averaged percentage of masks that will cause prediction change for each image (% windows per image).[6] As shown in Table 7, the overall average masked accuracy is high, and the percentage of images and windows for which the prediction changes is low. Such a small fraction of images with prediction changes enables us to achieve high provable robustness and maintain clean accuracy.

5.3.2 PatchGuard with Different Parameters

The receptive field size balances the trade-off between clean accuracy and provable robust accuracy of defended models. We report clean accuracy and provable robust accuracy of our defense with BagNet-33, BagNet-17, and BagNet-9, which have a receptive field of 33×33, 17×17, and 9×9, respectively, against different patch sizes in Table 8. 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 Mask-BN-33 has a higher clean accuracy than Mask-BN-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 Mask-BN-17. The trade-off between the clean accuracy and the robustness can be tuned with different receptive field sizes and should be carefully balanced when deploying the defense.

A large detection threshold improves clean accuracy but decreases provable robust accuracy of defended models. We study the model performance of BagNet-17 against a 2%

[6] We note that "% images" presented in Table 7 is an upper bound for our robust masking in Algorithm 1 because robust masking masks the window with the highest class evidence for each class while this analysis only removed wrong class evidence within the same window as the true class.

Table 10: Effect of feature types on Mask-BN-17

Patch size	1% pixels		2% pixels		3% pixels	
Accuracy	clean	robust	clean	robust	clean	robust
Logits	95.2%	89.0%	95.0%	86.7%	94.8%	83.0%
Con dence	87.9%	80.5%	87.9%	77.9%	88.0%	74.4%
Prediction	85.7%	77.3%	85.8%	74.1%	85.9%	70.3%

Table 11: Effect of feature types on Mask-DS

Patch size	1% pixels		2% pixels		3% pixels	
Accuracy	clean	robust	clean	robust	clean	robust
Logits	92.4%	76.9%	92.1%	68.9%	91.9%	61.6%
Con dence	92.3%	83.1%	92.1%	79.9%	92.1%	76.8%
Prediction	91.9%	82.5%	91.8%	79.4%	91.7%	76.4%

Table 12: Effect of over-conservatively large masks on Mask-BN-17

mask \ patch	clean	1% pixels	2% pixels	3% pixels
1% pixels	95.2%	89.0%	–	–
2% pixels	95.0%	88.2%	86.7%	–
3% pixels	94.8%	87.1%	85.3%	83.0%
4.5% pixels	94.6%	86.0%	84.1%	81.8%
CBN [59]	94.9%	74.6%	60.9%	45.9%
DS [28]	92.1%	82.3%	79.1%	75.7%

Table 13: Per-image inference time of different models

Model	ResNet-50	BagNet-17	DS-25-ResNet	Mask-BN	Mask-DS
Time	11.8ms	12.1ms	387.9ms	16.6ms	404.4ms

pixel patch as we change the detection thresh old T from 0.0 to 1.0. A threshold of zero means our detection will always return a suspicious window even if 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 Table 9. As we increase the detection thresh old T, we reduce the FP rate for clean images, at the cost of making it easier for an adversarial patch to succeed (Case IV (no suspicious window detected)). However, we note that false positives in the detection phase for clean images have a minimal impact on the clean accuracy because our models are generally invariant to feature masking, as already shown in Table 7. Thus, we nd T = 0 to be the best choice for this dataset (even with an FP of 100%); it results in the highest provable robust accuracy of 86.7% while only incurring a 0.5% clean accuracy drop compared to T = 1.

Different feature types greatly in uence the performance of defended models. In this analysis, we study the performance of the robust masking defense when using different types of features, namely logits, con dence values, and predictions. The results for Mask-BN-17 with different features are reported in Table 10. As shown in the table, using logits as the feature type has much better performance than con dence 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 con dence 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 11, Mask-DS works better when we use prediction or con dence 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 elds, and should be appropriately optimized to determine the best defense setting.

Over-conservatively large masks only have a small impact on defended models. PatchGuard's robust masking is deployed in a manner that is agnostic to the patch size by selecting a large mask window size that matches the upper bound of the patch size. In this analysis, we study the model performance when an over-conservatively large mask is used. Note that the provable robustness obtained with a larger mask for a larger patch can be directly applied to a smaller patch (e.g., an image that is robust against a 3% pixel patch is also robust against a 1% pixel patch). However, we can certify the robustness for more images when the actual patch size is smaller than the mask size (Appendix C).

We report the provable robust accuracy and clean accuracy of Mask-BN-17 with different patch sizes and attack-agnostic mask sizes in Table 12. First, robust masking with a larger mask can have a tighter provable robustness bound for a smaller patch. For example, when using a 3% pixel mask, the provable analysis in Algorithm 2 can only certify the robustness of 83.0% of test images for any patch size smaller than 3%. In contrast, the tighter provable analysis from Appendix C leads to a provable robust accuracy of 87.1% (4.1% improvement) for a 1% pixel patch. Second, over-conservatively using a larger mask size only leads to a slight drop in clean accuracy and provable robust accuracy. As we increase the mask size, the clean accuracy for 1% pixel patch only drops from 95.2% to 94.6% and the provable robust accuracy drops from 89.0% to 86.0%. We note that even when the mismatch is large (a 4.5% pixel mask for a 1% pixel patch), our defense still outperforms DS [28].

5.3.3 Defense Ef ciency

Robust masking only introduces a small defense overhead. In Table 13, we report the per-image inference time of different models on the ImageNette validation set. As shown in the table, the inference time of Mask-BN (16.6ms) is close to that of BagNet-17 (12.1ms). We have a similar observation for Mask-DS (404.4ms) and DS-25-ResNet (387.9ms). BagNet-like models (e.g., Mask-BN) are more ef cient than DS-like models (e.g., DS and Mask-DS). As discussed in Section 3.3, using an ensemble model (e.g., DS-ResNet) is

computationally expensive compared with using small convolution kernels in conventional CNNs (e.g., BagNet). From Table 13, we can see the inference time of BagNet-17 (12.1ms) much smaller than that of DS-25-ResNet (387.9ms). This difference leads to a huge efficiency gap between Mask-BN (16.6ms) and Mask-DS (404.4ms) as well as DS (387.9ms). Therefore, we suggest using small convolution kernels to build models with small receptive fields when the two approaches have similar defense performance.

6 Discussion

6.1 Generalization of Related Defenses

In this subsection, we will show that our defense framework is a generalization of other provably robust defenses such as Clipped BagNet [59], De-randomized Smoothing [28].

Clipped BagNet (CBN). CBN [59] proposes clipping the local logits tensor with function $\text{CLIP}(u) = \tanh(0.05 \cdot u) \cdot 1$ to improve the robustness of BagNet [5]. Since the range of $\tanh(\cdot)$ is bounded by $(-1; 1)$, the adversary can achieve at most $2k$ 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 analysis, CBN calculates the difference between the sum of unaffected logits values for the predicted class and the second predicted class as d, if $d > 2k$, 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 $T = 1$ (i.e., no detection), and adjust the clipping values c_l and c_h or the clipping function $\text{CLIP}(\cdot)$. Our evaluation shows that our defense significantly outperforms CBN across three different datasets. There are two major reasons for this performance difference: 1) CBN retains the malicious feature values while PatchGuard detects and masks them; 2) CBN uses conventional training while PatchGuard uses provable adversarial training (Appendix A).

De-randomized Smoothing (DS). DS [28] trains a 'smoothed' classifier on image pixel patches and computes the predicted class as the class with the majority vote among local predictions made from all pixel patches. 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 $2k$, where k is the upper bound of the number of corrupted predictions, DS certifies the robustness of the image. When we set the feature type to prediction and detection threshold to $T = 1$ (i.e., no detection), we can reduce Mask-DS to DS. Note that averaging all one-hot encoded local predictions gives the same global prediction as majority voting. The major cause of the relatively poor performance of DS is that its certification process discards the spatial information of each prediction while our robust masking defense utilizes the spatial information that all corrupted features are within a small window in the feature space.

We note that two defenses (BagCert [37] and Randomized Cropping [29]) appeared after the initial release of our paper preprint [55]; both of them can be regarded as instances of our PatchGuard framework, i.e., using CNNs with small receptive fields (modified BagNet [37]; image cropping [29]) and secure aggregation (majority voting [29, 37]). These two followup works further demonstrate the generality of PatchGuard.

6.2 Limitations and Future Work

While PatchGuard achieves state-of-the-art provable robustness and has higher or comparable clean accuracy compared with previous defenses, there is still a drop in clean accuracy compared with undefended models. We note that PatchGuard is compatible with any small-receptive-field CNN and secure aggregation mechanism, and we expect the trade-off between provable robustness and clean accuracy to be mitigated further given any progress in these two directions.

CNNs with small receptive fields. The use of small receptive fields provides substantial provable robustness but incurs a non-negligible clean accuracy drop for the two architectures (i.e., BagNet [5] and DS-ResNet [28]) used in this paper. In future work, we aim to explore better architectures and training methods for CNNs with small receptive fields in order to provide robustness against patch attacks while maintaining state-of-the-art clean accuracy. Any progress on this front will directly boost our defense performance since PatchGuard is compatible with any CNN with small receptive fields.

Secure feature aggregation. We presented robust masking to compute robust predictions from partially corrupted features. Robust masking works in a manner that is agnostic to the patch size by using a large mask, but a completely parameter-free defense may be more desirable. To this end, we observe that PatchGuard turns the problem of designing an adversarial patch defense into a robust aggregation problem, however, can we make a robust prediction from a partially corrupted feature tensor? Thus, techniques from robust statistics such as median, truncated mean, as well as differential privacy [14] can also be incorporated in our framework, some of which admit a parameter-free defense. We also plan to explore the design of custom secure aggregation mechanisms in future work that can further improve provable robustness.

7 Related Work

7.1 Localized Adversarial Perturbations

Most adversarial example research focuses on global L_p-norm bounded perturbations while localized adversaries have received much less attention. The adversarial patch attack was introduced by Brown et al. [6] 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 [22] aimed at inducing targeted misclassi cation in the digital domain.

Localized patch attacks against object detection [30, 51], semantic segmentation models [46] as well as training-time poisoning attacks using localized triggers [19, 31] have been proposed. Our threat model in this paper focuses on attacks against image classi cation models at test time; how to generalize our defense to the above settings can be an interesting future direction to study.

7.2 Adversarial Patch Defenses

Empirical defenses like Digital Watermark (DW) [20] and Local Gradient Smoothing (LGS) [39] were rst proposed to detect and neutralize adversarial patch. However, these heuristic defenses are vulnerable to adaptive attackers with knowledge of the defense.

Observing the ineffectiveness of DW and LGS, Chiang et al. [9] proposed the rst provable defense against adversarial patches via Interval Bound Propagation (IBP) [18, 38]. Despite its important theoretical contribution, the IBP defense has poor clean and provable robust accuracy, as shown in Table 4. Zhang et al. [59] proposed clipped BagNet (CBN) for provable robustness and Levine et al. [28] proposed building a `smoothed' classi er (DS) that outputs the class with the largest count from local predictions on all small pixel patches. We have shown that CBN and DS are instances of our general defense framework (Section 6.1), and PatchGuard has better performance due to the use of robust masking (Section 5.2). The Minority Report (MR) [34] defense was proposed in concurrent work, where the defender puts a mask at all possible locations and extracts patterns from model predictions. This defense can only provably detect an attack while PatchGuard also guarantees the recovery of the correct prediction. Moreover, MR performs masking in the image space which is computationally expensive and cannot scale to high-resolution images. However, if we can tolerate attack detection, MR has an advantage on low-resolution images (90.6% clean accuracy and 62.1% provable accuracy for 2.4%-pixel patch on CIFAR-10; compared to our 84.6% clean accuracy and 57.7% provable accuracy). How to extend PatchGuard for attack detection is an interesting direction of future work.

Another concurrent line of research has been on adversarial patch training [44, 54]. However, these works focus on empirical robustness and do not provide any provable guarantees.

7.3 Receptive Fields of CNNs

A number of papers have studied the in uence of the receptive eld [1, 5, 25, 32] on model performance in order to better understand the model behavior. BagNet [5] adopted the structure of ResNet-50 [21] but reduced the receptive eld size by replacing 3 3 kernels with 1 1 kernels. BagNet-17 can achieve similar top-5 validation accuracy as AlexNet [24] on ImageNet [12] dataset when each feature only looks at a 17 17 pixel region. The small receptive eld was used for better interpretability of model decisions in the original BagNet paper. In this work, we use the reduced receptive eld size to create models robust to adversarial patch attacks.

7.4 Other Adversarial Example Attacks and Defenses

The development of adversarial example-based attacks and defenses has been an extremely active research area over the past few years. Conventional adversarial attacks [8, 17, 41, 50] craft adversarial examples that have a small ℓ_p distance to clean examples but induce model misclassi cation. Many empirical defenses [35, 36, 42, 56] have been proposed to address the adversarial example vulnerability, but most of them can be easily bypassed by strong adaptive attackers [2, 7, 52]. The fragility of the empirical defenses has inspired provable or certi ed defenses [10, 18, 26, 38, 43, 53] as well as work on learning-theoretic bounds in the presence of adversaries [3, 11, 13, 45, 57]. In contrast, the focus of this paper is on localized adversarial patch attacks, and we refer interested readers to survey papers [40, 58] for a more detailed background on adversarial examples.

8 Conclusion

In this paper, we propose a general provable defense framework called PatchGuard that mitigates localized adversarial patch attacks. We identify large receptive elds and insecure aggregation mechanisms in conventional CNNs as the key sources of vulnerability to adversarial patches. To address these two problems, our defense proposes the use of models with small receptive elds to limit the number of features corrupted by the adversary which are then augmented with a robust masking defense to detect and mask the corrupted features to ensure secure feature aggregation. Our defense achieves state-of-the-art provable robust accuracy on ImageNet, ImageNette, and CIFAR-10 datasets. We hope that our general defense framework inspires further research to fully mitigate adversarial patch attacks.

Acknowledgements

We are grateful to David Wagner for shepherding the paper and anonymous reviewers at USENIX Security for their valuable feedback. This work was supported in part by the National Science Foundation under grants CNS-1553437 and CNS-1704105, the ARL's Army Arti cial Intelligence Innovation Institute (A2I2), the Of ce of Naval Research Young Investigator Award, the Army Research Of ce Young Investigator Prize, Faculty research award from Facebook, Schmidt DataX award, and Princeton E-f liates Award.

References

[1] Andre Araujo, Wade Norris, and Jack Sim. Computing receptive fields of convolutional neural networks. Distill, 2019. https://distill.pub/2019/computing-receptive-fields.

[2] Anish Athalye, Nicholas Carlini, and David 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), pages 274–283, 2018.

[3] Arjun Nitin Bhagoji, Daniel Cullina, and Prateek Mittal. Lower bounds on adversarial robustness from optimal transport. In Conference on Neural Information Processing Systems (NeurIPS), pages 7496–7508, 2019.

[4] Wieland Brendel. Pretrained bag-of-local-features neural networks. https://github.com/wielandbrendel/bag-of-local-features-models, 2020.

[5] Wieland Brendel and Matthias Bethge. Approximating CNNs with bag-of-local-features models works surprisingly well on ImageNet. In 7th International Conference on Learning Representations (ICLR), 2019.

[6] Tom B. Brown, Dandelion Mané, Aurko Roy, Martín Abadi, and Justin Gilmer. Adversarial patch. In Conference on Neural Information Processing Systems Workshops (NeurIPS Workshops), 2017.

[7] Nicholas Carlini and David A. Wagner. Adversarial examples are not easily detected: Bypassing ten detection methods. In Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security (AISec@CCS), pages 3–14, 2017.

[8] Nicholas Carlini and David A. Wagner. Towards evaluating the robustness of neural networks. In 2017 IEEE Symposium on Security and Privacy (S&P), pages 39–57, 2017.

[9] Ping-Yeh Chiang, Renkun Ni, Ahmed Abdelkader, Chen Zhu, Christoph Studor, and Tom Goldstein. Certified defenses for adversarial patches. In 8th International Conference on Learning Representations (ICLR), 2020.

[10] Jeremy M. Cohen, Elan Rosenfeld, and J. Zico Kolter. Certified adversarial robustness via randomized smoothing. In Proceedings of the 36th International Conference on Machine Learning (ICML), pages 1310–1320, 2019.

[11] Daniel Cullina, Arjun Nitin Bhagoji, and Prateek Mittal. PAC-learning in the presence of adversaries. In Conference on Neural Information Processing Systems (NeurIPS), pages 230–241, 2018.

[12] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Fei-Fei Li. ImageNet: A large-scale hierarchical image database. In 2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), pages 248–255, 2009.

[13] Elvis Dohmatob. Generalized no free lunch theorem for adversarial robustness. In Proceedings of the 36th International Conference on Machine Learning (ICML), pages 1646–1654, 2019.

[14] Cynthia Dwork and Aaron Roth. The algorithmic foundations of differential privacy. Foundations and Trends in Theoretical Computer Science, 9(3-4):211–407, 2014.

[15] Kevin Eykholt, Ivan Evtimov, Earlence Fernandes, Bo Li, Amir Rahmati, Chaowei Xiao, Atul Prakash, Tadayoshi Kohno, and Dawn Song. Robust physical-world attacks on deep learning visual classification. In 2018 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 1625–1634, 2018.

[16] fast.ai. ImageNette: A smaller subset of 10 easily classified classes from imagenet. https://github.com/fastai/imagenette, 2020.

[17] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In 3rd International Conference on Learning Representations (ICLR), 2015.

[18] Sven Gowal, Krishnamurthy Dvijotham, Robert Stanforth, Rudy Bunel, Chongli Qin, Jonathan Uesato, Relja Arandjelovic, Timothy Arthur Mann, and Pushmeet Kohli. Scalable verified training for provably robust image classification. In 2019 IEEE/CVF International Conference on Computer Vision (ICCV), pages 4841–4850, 2019.

[19] Tianyu Gu, Brendan Dolan-Gavitt, and Siddharth Garg. BadNets: Identifying vulnerabilities in the machine learning model supply chain. In Machine Learning and Computer Security Workshop (NeurIPS MLSec), 2017.

[20] Jamie Hayes. On visible adversarial perturbations & digital watermarking. In 2018 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPR Workshops), pages 1597–1604, 2018.

[21] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 770–778, 2016.

[22] Danny Karmon, Daniel Zoran, and Yoav Goldberg. LaVAN: Localized and visible adversarial noise. In Proceedings of the 35th International Conference on Machine Learning (ICML), pages 2512–2520, 2018.

[23] Alex Krizhevsky. Learning multiple layers of features from tiny images. https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf, 2009.

[24] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. ImageNet classification with deep convolutional neural networks. In Conference on Neural Information Processing Systems (NeurIPS), pages 1106–1114, 2012.

[25] Hung Le and Ali Borji. What are the receptive, effective receptive, and projective fields of neurons in convolutional neural networks? arXiv preprint arXiv:1705.07049, 2017.

[26] Mathias Lécuyer, Vaggelis Atlidakis, Roxana Geambasu, Daniel Hsu, and Suman Jana. Certified robustness to adversarial examples with differential privacy. In 2019 IEEE Symposium on Security and Privacy (S&P), pages 656–672, 2019.

[27] Alexander Levine and Soheil Feizi. Code for the paper "(de)randomized smoothing for certifiable defense against patch attacks". https://github.com/alevine0/patchSmoothing, 2020.

[28] Alexander Levine and Soheil Feizi. (De)randomized smoothing for certifiable defense against patch attacks. In Conference on Neural Information Processing Systems, (NeurIPS), 2020.

[29] Wan-Yi Lin, Fatemeh Sheikholeslami, jinghao shi, Leslie Rice, and J Zico Kolter. Certified robustness against physically-realizable patch attack via randomized cropping, 2021.

[30] Xin Liu, Huanrui Yang, Ziwei Liu, Linghao Song, Yiran Chen, and Hai Li. DPATCH: an adversarial patch attack on object detectors. In Workshop on Artificial Intelligence Safety 2019 co-located with the 33rd AAAI Conference on Artificial Intelligence 2019 (AAAI), volume 2301, 2019.

[31] Yingqi Liu, Shiqing Ma, Yousra Aafer, Wen-Chuan Lee, Juan Zhai, Weihang Wang, and Xiangyu Zhang. Trojaning attack on neural networks. In 25th Annual Network and Distributed System Security Symposium (NDSS) 2018.

[32] Wenjie Luo, Yujia Li, Raquel Urtasun, and Richard S. Zemel. Understanding the effective receptive field in deep convolutional neural networks. In Conference on Neural Information Processing Systems (NeurIPS) pages 4898–4906, 2016.

[33] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In 6th International Conference on Learning Representations (ICLR), 2018.

[34] Michael McCoyd, Won Park, Steven Chen, Neil Shah, Ryan Roggenkemper, Minjune Hwang, Jason Xinyu Liu, and David A. Wagner. Minority reports defense: Defending against adversarial patches. In Applied Cryptography and Network Security Workshops (ACNS Workshops), volume 12418, pages 564–582. Springer, 2020.

[35] Dongyu Meng and Hao Chen. Magnet: A two-pronged defense against adversarial examples. In Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security (CCS), pages 135–147, 2017.

[36] Jan Hendrik Metzen, Tim Genewein, Volker Fischer, and Bastian Bischoff. On detecting adversarial perturbations. In 5th International Conference on Learning Representations (ICLR), 2017.

[37] Jan Hendrik Metzen and Maksym Yatsura. Efficient certified defenses against patch attacks on image classifiers. In 9th International Conference on Learning Representations (ICLR), 2021.

[38] Matthew Mirman, Timon Gehr, and Martin T. Vechev. Differentiable abstract interpretation for provably robust neural networks. In Proceedings of the 35th International Conference on Machine Learning (ICML), pages 3575–3583, 2018.

[39] Muzammal Naseer, Salman Khan, and Fatih Porikli. Local gradients smoothing: Defense against localized adversarial attacks. In IEEE Winter Conference on Applications of Computer Vision (WACV), pages 1300–1307, 2019.

[40] Nicolas Papernot, Patrick McDaniel, Arunesh Sinha, and Michael P Wellman. Sok: Security and privacy in machine learning. In 2018 IEEE European Symposium on Security and Privacy (EuroS&P), pages 399–414, 2018.

[41] Nicolas Papernot, Patrick D. McDaniel, Somesh Jha, Matt Fredrikson, Z. Berkay Celik, and Ananthram Swami. The limitations of deep learning in adversarial settings. In IEEE European Symposium on Security and Privacy (EuroS&P), pages 372–387, 2016.

[42] Nicolas Papernot, Patrick D. McDaniel, Xi Wu, Somesh Jha, and Ananthram Swami. Distillation as a defense to adversarial perturbations against deep neural networks. In IEEE Symposium on Security and Privacy (S&P) pages 582–597, 2016.

[43] Aditi Raghunathan, Jacob Steinhardt, and Percy Liang. Certified defenses against adversarial examples. In 6th International Conference on Learning Representations (ICLR), 2018.

[44] Sukrut Rao, David Stutz, and Bernt Schiele. Adversarial training against location-optimized adversarial patches. In *European Conference on Computer Vision Workshops (ECCV Workshops)*, 2020.

[45] Ludwig Schmidt, Shibani Santurkar, Dimitris Tsipras, Kunal Talwar, and Aleksander Madry. Adversarially robust generalization requires more data. In *Advances in Neural Information Processing Systems*, pages 5014–5026, 2018.

[46] Vikash Sehwag, Chawin Sitawarin, Arjun Nitin Bhagoji, Arsalan Mosenia, Mung Chiang, and Prateek Mittal. Not all pixels are born equal: An analysis of evasion attacks under locality constraints. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security (CCS)*, pages 2285–2287, 2018.

[47] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *3rd International Conference on Learning Representations (ICLR)*, 2015.

[48] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander A. Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI)*, pages 4278–4284, 2017.

[49] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–9, 2015.

[50] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian J. Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *2nd International Conference on Learning Representations (ICLR)*, 2014.

[51] Simen Thys, Wiebe Van Ranst, and Toon Goedemé. Fooling automated surveillance cameras: Adversarial patches to attack person detection. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPR Workshops)*, pages 49–55, 2019.

[52] Florian Tramer, Nicholas Carlini, Wieland Brendel, and Aleksander Madry. On adaptive attacks to adversarial example defenses. In *2020 USENIX Security and AI Networking Summit (ScAINet)*, 2020.

[53] Eric Wong and J. Zico Kolter. Provable defenses against adversarial examples via the convex outer adversarial polytope. In *Proceedings of the 35th International Conference on Machine Learning (ICML)*, pages 5283–5292, 2018.

[54] Tong Wu, Liang Tong, and Yevgeniy Vorobeychik. Defending against physically realizable attacks on image classification. In *8th International Conference on Learning Representations (ICLR)*, 2020.

[55] Chong Xiang, Arjun Nitin Bhagoji, Vikash Sehwag, and Prateek Mittal. Patchguard: Provable defense against adversarial patches using masks on small receptive fields. *arXiv preprint arXiv:2005.10884*, 2020.

[56] Weilin Xu, David Evans, and Yanjun Qi. Feature squeezing: Detecting adversarial examples in deep neural networks. In *25th Annual Network and Distributed System Security Symposium (NDSS)*, 2018.

[57] Dong Yin, Ramchandran Kannan, and Peter Bartlett. Rademacher complexity for adversarially robust generalization. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*, pages 7085–7094, 2019.

[58] Xiaoyong Yuan, Pan He, Qile Zhu, and Xiaolin Li. Adversarial examples: Attacks and defenses for deep learning. *IEEE Transactions on Neural Networks and Learning Systems*, 30(9):2805–2824, 2019.

[59] Zhanyuan Zhang, Benson Yuan, Michael McCoyd, and David Wagner. Clipped bagnet: Defending against sticker attacks with clipped bag-of-features. In *3rd Deep Learning and Security Workshop (DLS)*, 2020.

A Provable Adversarial Training

In order to improve the provable robust accuracy, we train a BagNet with a mask over the region with the largest true class evidence. This training mimics the procedure of our provable analysis in Algorithm 2, and we call it *provable adversarial training*. In Table 14, we report the results for Mask-BN-17 with and without provable adversarial training against a 2% pixel patch on ImageNet/ImageNette and a 2.4% pixel patch on CIFAR-10. We can see from the table that provable adversarial training significantly improves provable robustness. We note that we do not do provable adversarial training for DS-ResNet because it is too expensive to computing its all local features during the training. Further details of model training are available in our technical report [55].

Table 14: Effect of provable adversarial training on Mask-BN-17

Dataset	ImageNet		ImageNette		CIFAR-10	
Accuracy	clean	robust	clean	robust	clean	robust
Conventional training	54.4%	13.3%	93.9%	83.8%	82.6%	31.7%
Provable adv. training	54.6%	26.0%	95.0%	86.7%	83.9%	47.3%

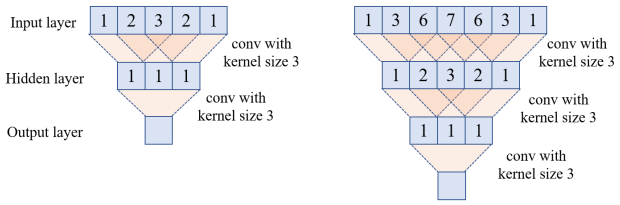

Figure 6: Toy example of 1-D convolution computation

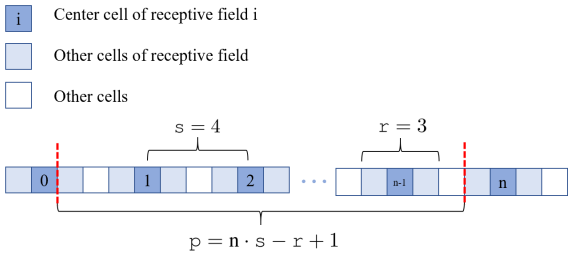

Figure 7: Example of computing window size.

B Details of Receptive Fields

Local features focus on the center of the receptive field. In Section 3.1, we mentioned that a particular local feature focuses exponentially more on the center of its receptive field. We provide the intuition for this argument in Figure 6. The left part of the figure illustrates a 1-D example of convolution computation in which the input has five cells and will go through two convolution layers with a kernel size of 3 to compute the final output. Each cell in the hidden layer (i.e., the output of the first convolution layer) looks at 3 input cells, and the output cell looks at three hidden cells. We count the number of times each cell is looked at when computing the output cell and plot it in the figure. As we can see, the center cell of the input layer receives the most attention (being looked at 3 times). Moreover, as the number of layers increases (a similar example for 3 convolution layers is plotted in the right part of Figure 6), the difference in attention between the center cell and the rightmost/leftmost cell will increase exponentially. Therefore, a particular feature focuses exponentially more on the center of its receptive field, and an adversary controlling the center cell will have a larger capacity to manipulate the final output features.

Computing the Window Size. One crucial step of our robust masking defense is to determine the window size, and we show in Section 3.4 and Equation 1 that the window size w can be computed as $w = d(p + r - 1)/se$, where p is the upper bound on the patch size, r is the size of receptive field, and s is the stride of receptive field. In Figure 7, we provide the intuition for Equation 1. In this example, we assume the stride $s = 4$, and the size of the receptive field $r = 3$. We distinguish the centers of receptive fields, the other cells in the receptive fields, and the other cells with different colors. Note that we choose a large stride s such that adjacent receptive fields do not overlap for a better visual demonstration; the derived equation is applicable to smaller s or larger r. In Figure 7, we want to determine the largest patch size p such that the patch only appears in n but not n+1 receptive fields. We plot the boundary of the largest patch with red dash line in the figure. The left part of the patch covers the rightmost cells of receptive field 0, and the right part does not appear in receptive field n. Based on Figure 7, we can compute $p = n \cdot s - r + 1$. Next, we can substitute n with w, use $d\,e$ for generalization to any patch size, and finally get $w = d(p + r - 1)/se$. We note that the network architectures [4, 27] used in this paper have $s = 8$ for BagNet and $s = 1$ for DS-ResNet.

Table 15: Top-k accuracies of Mask-BN-17 on ImageNet

Patch size	1% pixels		2% pixels		3% pixels	
Accuracy	clean	robust	clean	robust	clean	robust
Top-1	55.1%	32.2%	54.6%	26.0%	54.1%	19.7%
Top-2	65.9%	48.3%	65.5%	43.8%	64.9%	38.2%
Top-3	71.3%	52.2%	70.8%	48.7%	70.2%	44.1%
Top-4	74.6%	53.9%	74.2%	51.3%	73.7%	47.4%
Top-5	77.0%	54.8%	76.6%	52.9%	76.2%	49.6%

C Tighter Provable Analysis for Over-conservative Mask Size

Recall that the mask window size is a tunable security parameter. Robust masking can prove robustness for any patch is smaller than the mask. In this section, we discuss a tighter version of Lemma 1 when the defender overestimates the worst-case patch size and use a larger mask window size. Let W be the set of all possible malicious windows and V be the set of all possible detected windows whose sizes are larger than malicious windows. We can have the following generalized Lemma.

Lemma 2. *Given a malicious window* $\mathbf{w} \in W$*, a class* $\bar{y} \in Y$*, and the set of all possible detected windows* V*, the clipped and masked class evidence of class* $\bar{y}$ *(i.e.,* $s_{\bar{y}}$*) can be no larger than* $\text{SUM}(\hat{\mathbf{u}}_{\bar{y}} \odot (\mathbf{1} - \mathbf{v}_{\mathbf{w}}))/(1 - T)$*, where* $\mathbf{v}_{\mathbf{w}} = \arg\max_{\mathbf{v} \in V_{\mathbf{w}}} \text{SUM}(\hat{\mathbf{u}}_{\bar{y}} \odot \mathbf{v})$ *and* $V_{\mathbf{w}} = \{\mathbf{v} \in V | \text{SUM}(\mathbf{w} \odot \mathbf{v}) = \text{SUM}(\mathbf{w})\}$.

In this lemma, $V_{\mathbf{w}}$ is the set of possible mask windows that cover the entire malicious window $\mathbf{w}$, and $\mathbf{v}_{\mathbf{w}}$ is the mask window in $V_{\mathbf{w}}$ with the largest class evidence. This bound reduces to the bound of Lemma 1 when the sizes of malicious window and mask window (i.e., the output of subprocedure DETECT) are the same: $V_{\mathbf{w}} = \{\mathbf{w}\}$ and thus $\mathbf{v}_{\mathbf{w}} = \mathbf{w}$. The proof of Lemma 2 is in the same spirit as that of Lemma 1 and is available in our technical report [55].