Certified Patch Robustness via Smoothed Vision Transformers

Hadi Salman* MIT hady@mit.edu
Saachi Jain* MIT saachij@mit.edu
Eric Wong* MIT wongeric@mit.edu
Aleksander Mądry MIT madry@mit.edu

Abstract

Certified patch defenses can guarantee robustness of an image classifier to arbitrary changes within a bounded contiguous region. But, currently, this robustness comes at a cost of degraded standard accuracies and slower inference times. We demonstrate how using vision transformers enables significantly better certified patch robustness that is also more computationally efficient and does not incur a substantial drop in standard accuracy. These improvements stem from the inherent ability of the vision transformer to gracefully handle largely masked images.[1]

1 Introduction

High-stakes scenarios warrant the development of certifiably robust models that are *guaranteed* to be robust to a set of transformations. These techniques are beginning to find applications in real-world settings, such as verifying that aircraft controllers behave safely in the presence of approaching airplanes [JK19], and ensuring the stability of automotive systems to sensor noise [Won+20].

We study robustness in the context of adversarial patches—a broad class of arbitrary changes contained within a small, contiguous region. Adversarial patches capture the essence of a range of maliciously designed physical objects such as adversarial glasses [Sha+16], stickers/graffiti [Evt+18], and clothing [Wu+20]. Researchers have used adversarial patches to fool image classifiers [Bro+18], manipulate object detectors [LK19; Hoo+20], and disrupt optical flow estimation [Ran+19].

Adversarial patch defenses can be tricky to evaluate—recent work broke several empirical defenses [BMV18; Hay18; NKP19] with stronger adaptive attacks [Tra+20; Chi+20]. This motivated *certified* defenses, which deliver provably robust models without having to rely on an empirical evaluation. However, certified guarantees tend to be modest and come at a cost: poor standard accuracy and slower inference times [LF20b; LF20a; Zha+20; Xia+21]. For example, a top-performing, recently proposed method reduces standard accuracy by 30% and increases inference time by two orders of magnitude, while certifying only 13.9% robust accuracy on ImageNet against patches that take up 2% of the image [LF20a]. These drawbacks are commonly accepted as the cost of certification, but severely limit the applicability of certified defenses. Does certified robustness really need to come at such a high price?

Our contributions

In this paper, we demonstrate how to leverage vision transformers (ViTs) [Dos+21] to create certified patch defenses that achieve significantly higher robustness guarantees than prior work. Moreover, we show that certified patch defenses with ViTs can actually maintain standard accuracy and inference times comparable to standard (non-robust) models. At its core, our methodology exploits the token-based nature of attention modules used in ViTs to gracefully handle the ablated images used in certified patch defenses. Specifically, we demonstrate the following:

*Equal contribution.
[1]Our code is available at https://github.com/MadryLab/smoothed-vit.

Improved guarantees via smoothed vision transformers. We find that using ViTs as the backbone of the derandomized smoothing defense [LF20a] enables significantly improved certified patch robustness. Indeed, this change alone boosts certified accuracy by up to 13% on ImageNet, and 5% on CIFAR-10 over similarly sized ResNets.

Standard accuracy comparable to that of standard architecures. We demonstrate that ViTs enable certified defenses with standard accuracies comparable to that of standard, non-robust models. In particular, our largest ViT improves state-of-the-art certified robustness on ImageNet while maintaining standard accuracy that is similar to that of a non-robust ResNet (>70%).

Faster inference. We modify the ViT architecture to drop unnecessary tokens, and reduce the smoothing process to pass over mostly redundant computation. These changes turn out to vastly speed up inference time for our smoothed ViTs. In our framework, a forward pass on ImageNet becomes up to two orders of magnitude faster than that of prior certified defenses, and is close in speed to a standard (non-robust) ResNet.

2 Certified patch defense with smoothing & transformers

Smoothing methods are a general class of certified defenses that combine the predictions of a classifier over many variations of an input to create predictions that are certifiably robust [CRK19; LF20b]. One such method that obtains robustness to adversarial patches is derandomized smoothing [LF20a], which aggregates a classifier's predictions on various *image ablations* that mask most of the image out.

These approaches typically use CNNs, a common default model for computer vision tasks, to evaluate the image ablations. The starting point of our approach is to ask: are convolutional architectures the right tool for this task? The crux of our methodology is to leverage vision transformers, which we demonstrate are more capable of gracefully handling the image ablations that arise in derandomized smoothing.

2.1 Preliminaries

Image ablations. Image ablations are variations of an image where all but a small portion of the image is masked out [LF20a]. For example, a column ablation masks the entire image except for a column of a fixed width (see Figure 1 for an example). We focus primarily on column ablations and explore the more general block ablation in Appendix E.

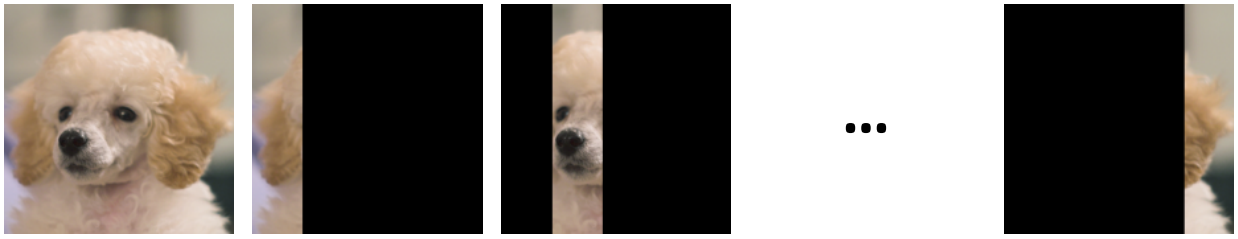


Figure 1: Examples of column ablations for the left-most image with column width 19px.

For a input $h \times w$ sized image $\mathbf{x}$, we denote by $\mathcal{S}_b(\mathbf{x})$ the set of all possible column ablations of width b. A column ablation can start at any position and wrap around the image, so there are w total ablations in $\mathcal{S}_b(\mathbf{x})$.

Derandomized smoothing. Derandomized smoothing [LF20a] is a popular approach for certified patch defenses that constructs a *smoothed classifier* comprising of two main components: (1) a *base classifier*, and (2) a set of image ablations used to smooth the base classifier. Then, the resulting smoothed classifier returns

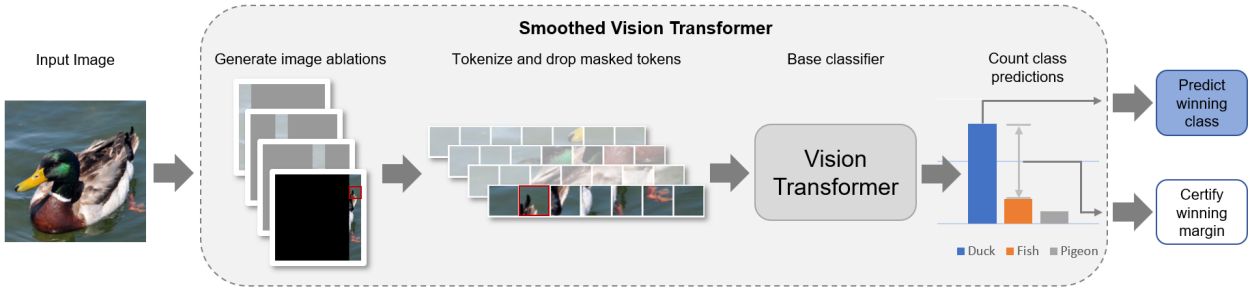


Figure 2: Illustration of the smoothed vision transformer. For a given image, we first generate a set of ablations. We encode each ablation into tokens, and drop fully masked tokens. The remaining tokens for each ablation are then fed into a vision transformer, which predicts a class label for each ablation. We predict the class with the most predictions over all the ablations, and use the margin to the second-place class for robustness certification.

the most frequent prediction of the base classifier over the ablation set $\mathcal{S}_b(\mathbf{x})$. Specifically, for an input image $\mathbf{x}$, ablation set $\mathcal{S}_b(\mathbf{x})$, and a base classifier f, a smoothed classifier g is defined as:

$$g(\mathbf{x}) = \arg\max_c n_c(\mathbf{x}) \tag{1}$$

where

$$n_c(\mathbf{x}) = \sum_{\mathbf{x}' \in S_b(\mathbf{x})} \mathbb{I}\{f(\mathbf{x}') = c\}$$

denotes the number of image ablations that were classified as class c. We refer to the fraction of images that the smoothed classifier correctly classifies as *standard accuracy*.

A smoothed classifier is *certifiably robust* for an input image if the number of ablations for the most frequent class exceeds the second most frequent class by a large enough margin. Intuitively, a large margin makes it impossible for an adversarial patch to change the prediction of a smoothed classifier since a patch can only affect a limited number of ablations.

Specifically, let Δ be the maximum number of ablations in the ablation set $S_b(\mathbf{x})$ that an adversarial patch can simultaneously intersect (e.g., for column ablations of size b, an $m \times m$ patch can intersect with at most $\Delta = m + b - 1$ ablations). Then, a smoothed classifier is certifiably robust on an input $\mathbf{x}$ if it is the case that for the predicted class c:

$$n_c(\mathbf{x}) > \max_{c' \neq c} n_{c'}(\mathbf{x}) + 2\Delta. \tag{2}$$

If this threshold is met, the most frequent class is guaranteed to not change even if an adversarial patch compromises every ablation it intersects. We denote the fraction of predictions by the smooth classifier that are both correct and certifiably robust (according to Equation 2) as *certified accuracy*.

Vision transformers. A key component of our approach is the vision transformer (ViT) architecture [Dos+21]. In contrast to convolutional architecures, ViTs use self-attention layers instead of convolutional layers as their primary building block and are inspired by the success of self-attention in natural language processing [Vas+17]. ViTs process images in three main stages:

1. *Tokenization:* The ViTs split the image into $p \times p$ patches. Each patch is then embedded into a positionally encoded *token*.
2. *Self-Attention:* The set of tokens are then passed through a series of multi-headed self-attention layers [Vas+17].
3. *Classification head:* The resulting representation is fed into a fully connected layer to make predictions for classification.

Table 1: Summary of our ImageNet results and comparisons to certified patch defenses from the literature: Clipped Bagnet (CBG), Derandomized Smoothing (DS), and PatchGuard (PG). Time refers to the inference time for a batch of 1024 images, b is the ablation size, and s is the ablation stride. An extended version is in Appendix F.

Standard and Certified Accuracy on ImageNet (%)					
	Standard	1% pixels	2% pixels	3% pixels	Time (sec)
Baselines					
Standard ResNet-50	76.1	—	—	—	0.67
WRN-101-2	78.85	—	—	—	3.1
ViT-S	79.90	—	—	—	0.4
ViT-B	81.80	—	—	—	0.95
CBN [Zha+20]	49.5	13.4	7.1	3.1	3.05
DS [LF20a]*	44.4	17.7	14.0	11.2	149.5
PG [Xia+21]†	55.1†	32.3†	26.0†	19.7†	3.05
Smoothed models					
ResNet-50 (b = 19)	51.5	22.8	18.3	15.3	149.5
ViT-S (b = 19)	**63.5**	**36.8**	**31.6**	**27.9**	**14.0**
WRN-101-2 (b = 19)	61.4	33.3	28.1	24.1	694.5
ViT-B (b = 19)	69.3	**43.8**	**38.3**	**34.3**	31.5
ViT-B (b = 37)	**73.2**	43.0	38.2	34.1	58.7
ViT-B (b = 19, s = 10)	68.3	36.9	36.9	31.4	**3.2**

2.2 Smoothed vision transformers

Two central properties of vision transformers make ViTs particularly appealing for processing the image ablations that arise in derandomized smoothing. Firstly, unlike CNNs, ViTs process images as sets of tokens. ViTs thus have the natural capability to simply drop unnecessary tokens from the input and "ignore" large regions of the image, which can greatly speed up the processing of image ablations.

Moreover, unlike convolutions which operate locally, the self-attention mechanism in ViTs shares information *globally* at every layer [Vas+17]. Thus, one would expect ViTs to be better suited for classifying image ablations, as they can dynamically attend to the small, unmasked region. In contrast, a CNN must gradually build up its receptive field over multiple layers and process masked-out pixels.

Guided by these intuitions, our methodology leverages the ViT architecture as the base classifier for processing the image ablations used in derandomized smoothing. We first demonstrate that these *smoothed vision transformers* enable substantially improved robustness guarantees, without losing much standard accuracy (Section 3). We then modify the ViT architecture and smoothing procedure to drastically speed up the cost of inference of a smoothed ViT (Section 4). We present an overview of our approach in Figure 2.

Setup. We focus primarily on the column smoothing setting and defer block smoothing results to Appendix E. We consider the CIFAR-10 [Kri09] and ImageNet [Den+09] datasets, and perform our analysis on three sizes of vision transformers—ViT-Tiny (ViT-T), ViT-Small (ViT-S), and ViT-Base (ViT-B) models [Wig19; Dos+21]. We compare to residual networks of similar size—ResNet-18, ResNet-50 [He+16], and Wide ResNet-101-2 [ZK16], respectively. Further details of our experimental setup are in Appendix A.

*We found that ResNets could achieve a significantly higher certified accuracy than was reported by Levine and Feizi [LF20a] if we use early stopping-based model selection. We elaborate further in Appendix A.

†The PatchGuard defense uses a specific mask size that guarantees robustness to patches smaller than the mask, and provides no guarantees for larger patches. In this table, we report their best results: each patch size corresponds to a separate model that achieves 0% certified accuracy against larger patches. Comparisons across the individual models can be found in Appendix F.

Table 2: Summary of our CIFAR-10 results and comparisons to certified patch defenses from the literature: Clipped Bagnet (CBG), Derandomized Smoothing (DS), and PatchGuard (PG). Here, b is the column ablation size out of 32 pixels. An extended version is in Appendix F.

Standard and Certified Accuracy on CIFAR-10 (%)			
	Standard	2×2	4×4
Baselines			
CBN [Zha+20]	84.2	44.2	9.3
DS [LF20a]*	83.9	68.9	56.2
PG [Xia+21]†	84.7†	69.2†	57.7†
Smoothed models			
ResNet-50 (b = 4)	86.4	71.6	59.0
ViT-S (b = 4)	**88.4**	**75.0**	**63.8**
WRN-101-2 (b = 4)	88.2	73.9	62.0
ViT-B (b = 4)	**90.8**	**78.1**	**67.6**

3 Improving certified and standard accuracies with ViTs

Recall that even though certified patch defenses can guarantee robustness to patch attacks, this robustness typically does not come for free. Indeed, certified patch defenses tend to have substantially lower standard accuracy when compared to typical (non-robust) models, while delivering a fairly limited degree of (certified) robustness.

In this section, we show how to use ViTs to substantially improve both standard and certified accuracies for certified patch defenses. To this end, we first empirically demonstrate that ViTs are a more suitable architecture than traditional convolutional networks for classifying the image ablations used in derandomized smoothing (Section 3.1). Specifically, this change in architecture alone yields models with significantly improved standard and certified accuracies. We then show how a careful selection of smoothing parameters can enable smoothed ViTs to have even higher standard accuracies that are comparable to typical (non-robust) models, without sacrificing much certified performance (Section 3.2).

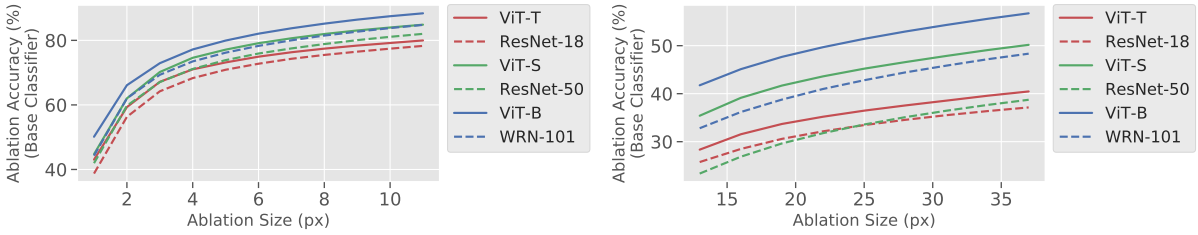
Our ImageNet and CIFAR-10 results are summarized in Table 1 and Table 2, respectively. We further include the inference time to evaluate a batch of images, using the modifications described in Section 4. See Appendix F for extended tables covering a wider range of experiments.

3.1 ViTs outperform ResNets on image ablations.

We first isolate the effect of using a ViT instead of a ResNet as the base classifier for derandomized smoothing. Specifically, we keep all smoothing parameters fixed and only vary the base classifier. Following Levine and Feizi [LF20a], we use column ablations of width $b = 4$ for CIFAR-10 and $b = 19$ for ImageNet for both training and certification.

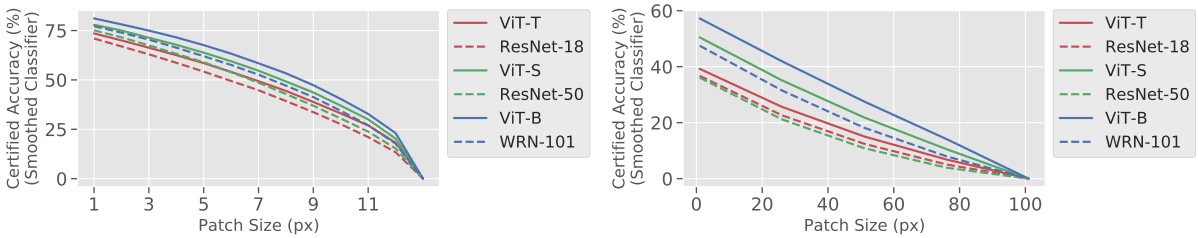
Ablation accuracy. The performance of derandomized smoothing entirely depends on whether the base classifier can accurately classify ablated images. We thus measure the accuracy of ViTs and ResNets at classifying column ablated images across a range of evaluation ablation sizes as shown in Figure 3. We find that ViTs are significantly more accurate on these ablations than comparably sized ResNets. For example, on ImageNet, ViT-S has up to 12% higher accuracy on ablations than ResNet-50.

Certified patch robustness. We next measure the effect of improved ablation accuracy on certified accuracy. We find that using a ViT as the base classifier in derandomized smoothing substantially boosts certified accuracy compared to ResNets across a range of model sizes and adversarial patch sizes, as shown in Figure 4. For example, against 32×32 adversarial patches on ImageNet (2% of the image), a smoothed



(a) CIFAR-10 (b) ImageNet

Figure 3: Accuracies on column-ablated images for models on CIFAR-10 and ImageNet. The models were trained on column ablations of width $b = 19$ for ImageNet and $b = 4$ for CIFAR-10, and evaluated on a range of ablation sizes. ViTs outperform ResNets on image ablations by a sizeable margin.



(a) CIFAR-10 (b) ImageNet

Figure 4: Certified accuracies for ViT and ResNet models on CIFAR-10 and ImageNet for various adversarial patch sizes. Certification was performed using a fixed ablation of size $b = 4$ for CIFAR-10 and $b = 19$ for ImageNet (as in [LF20a]).

ViT-S improves certified accuracy by 14% over a smoothed ResNet-50, while the larger ViT-B reaches a certified accuracy of 39%—well above the highest reported baseline of 26% [Xia+21][2].

Standard accuracy. We further find that smoothed ViTs can mitigate the precipitous drop in standard accuracy observed in previously proposed certified defenses, particularly so for larger architectures and datasets. Indeed, the smoothed ViT-B remains 69% accurate on ImageNet—14.2% higher standard accuracy than that of the best performing prior work (Table 1). A full comparison between the performance of smoothed models and their non-robust counterparts can be found in Appendix F.

3.2 Ablation size matters

In the previous section, we fixed the width of column ablations at $b = 19$ for derandomized smoothing on ImageNet, following [LF20a]. We now demonstrate that properly choosing the ablation size can improve the standard accuracy even further—by 4% on ImageNet—without sacrificing certified performance.

Specifically, we take ImageNet models trained on column ablations with width $b = 19$, and change the smoothing procedure to use a different width at *test* time. We report the resulting standard and certified accuracies in Figure 5, and defer additional experiments on changing the ablation size during training to Appendix B.1.

Although Levine and Feizi [LF20a] found a steep trade-off between certified and standard accuracy in CIFAR-10 (which we verify in Appendix B.2), we find this to not be the case for ImageNet for either CNNs or ViTs. We can thus substantially increase the ablation size to improve standard accuracy *without* significantly dropping certified performance as shown in Figure 5. For example, increasing the width of

[2]The highest reported certified accuracy in the literature for this patch size on ImageNet is 26% from PatchGuard [Xia+21]. However, this defense uses a masking technique that is optimized for this particular patch size, and achieves 0% certified accuracy against larger patches.

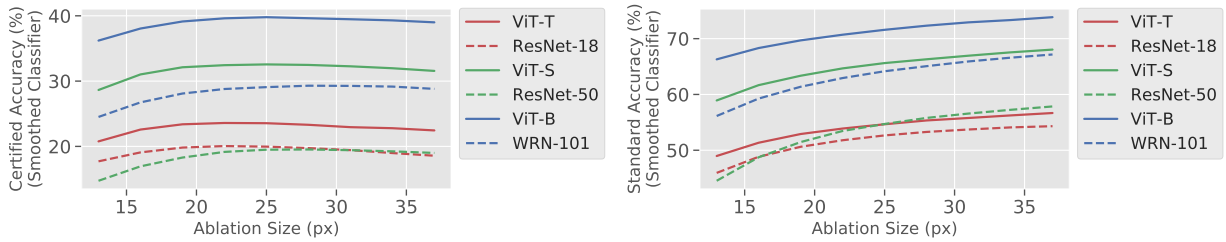


Figure 5: Certified (left) and standard (right) accuracies for a collection of smoothed models trained with a fixed ablation size $b = 19$ on ImageNet, and evaluated with varying ablation sizes. Certified accuracy remains stable across a range of ablation sizes, while standard accuracy substantially improves with larger ablations.

column ablations to $b = 37$ improves the standard accuracy of the smoothed ViT-B model by nearly 4% to 73% while maintaining a 38% certified accuracy against 32×32 patches. In addition to being 12% higher than the standard accuracy of the best performing prior work, this model's standard accuracy is only 3% lower than that of a *non-robust* ResNet-50.

Thus, using smoothed ViTs, we can achieve state-of-the-art certified robustness to patch attacks in the ImageNet setting while attaining standard accuracies that are more comparable to those of non-robust ResNets.

4 Faster inference with ViTs

Derandomized smoothing with column ablations is an expensive operation, especially for large images. Indeed, an image with $h \times w$ pixels has w column ablations, so the forward pass of smoothed model is w times slower than a normal forward pass—*two orders of magnitude* slower on ImageNet.

To address this, we first modify the ViT architecture to avoid unnecessary computation on masked pixels (Section 4.1). We then demonstrate that reducing the number of ablations via striding offers further speed up (Section 4.2). These two (complementary) modifications vastly improve the inference time for smoothed ViTs, making them comparable in speed to standard (non-robust) convolutional architectures.

4.1 Dropping masked tokens

Recall that the first operation in a ViT is to split and encode the input image as a set of *tokens*, where each token corresponds to a patch in the image. However, for image ablations, a large number of these tokens correspond to fully masked regions of the image.

Our strategy is to pass only the *subset* of tokens that contain an unmasked part of the original image, thus avoiding computation on fully masked tokens. Specifically, given an image ablation, we alter the ViT architecture to do the following steps:

1. Positionally encode the entire ablated image into a set of tokens.
2. Drop any tokens that correspond to a *fully* masked region of the input.
3. Pass the remaining tokens through the self-attention layers.

As one would expect, since the positional encoding maintains the spatial information of the remaining tokens, the ViT's accuracy on image ablations barely changes when we drop the fully masked tokens. We defer a detailed analysis of this phenomenon, along with a formal description of the token-dropping procedure to Appendix C.

Table 3: Multiplicative speed up of inference for a smoothed ViT with dropped tokens over a smoothed ResNet, measured over a batch of 1024 images with $b = 19$.

	ResNet-18	ResNet-50	WRN-101
ViT-T	**5.85x**	21.96x	101.99x
ViT-S	2.85x	**10.68x**	49.62x
ViT-B	1.26x	4.75x	**22.04x**

Computational complexity. We now provide an informal summary of the computational complexity of this procedure, and defer a formal asymptotic analysis to Appendix C.1. After tokenization, the bulk of a ViT consists of two main operation types:

- *Attention operators*, which have costs that scale quadratically with the number of tokens but linearly in the hidden dimension.
- *Fully-connected operators*, which have costs that scale linearly with the number of tokens but quadratically in the hidden dimension.

Reducing the number of tokens thus directly reduces the cost of attention and fully connected operators at a quadratic and linear rate, respectively. For a small number of tokens, the linear scaling from the fully-connected operators tends to dominate. The cost of processing column ablations thus scales linearly with the width of the column, which we empirically validate in Figure 6. Further details about how we time these models can be found in Appendix A.4.

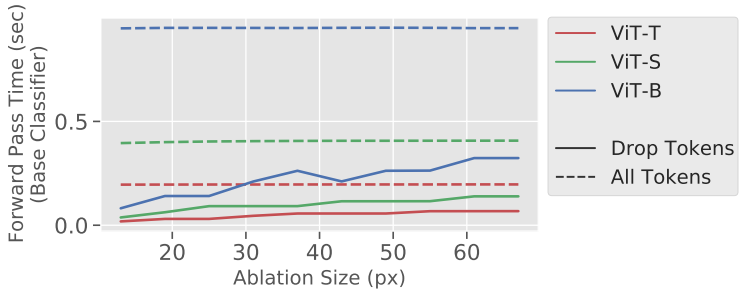


Figure 6: The average time to compute a forward pass for ViTs on 1024 column ablated images with varying ablation sizes, with and without dropping masked tokens. The cost of processing a full image without dropping masked tokens corresponds to the maximum ablation size $b = 224$.

4.2 Empirical speed-up for smoothed ViTs

Smoothed classifiers must process a large number of image ablations in order to make predictions and certify robustness. Consequently, using our ViT (with dropped tokens) as the base classifier for derandomized smoothing directly speeds up inference time. In this section, we explore how much faster smoothed ViTs are in practice.

We first measure the number of images per second that smoothed ViTs and smoothed ResNets can process. We use column ablations of size $b = 19$ on ImageNet, following Levine and Feizi [LF20a]. In Table 3 that describes our results, we find speedups of 5-22x for smoothed ViTs over smoothed ResNets of similar size, with larger architectures showing greater gains. Notably, using our largest ViT (ViT-B) as the base classifier is 1.25x faster than using a ResNet-18, despite being 8x larger in parameter count. Dropping masked tokens thus substantially speeds up inference time for smoothed ViTs, to the point where using a large ViT is comparable in speed to using a small ResNet.

Strided ablations. We now consider a complementary means of speeding up smoothed classifiers: directly reducing the size of the ablation set via *strided* ablations. Specifically, instead of using every possible ablation, we can subsample every s-th ablation for a given stride s. Striding can reduce the total number of ablations (and consequently speed up inference) by a factor of s, *without* substantially hurting standard or certified accuracy (Table 1). We study this in more detail in Appendix D.

Strided ablations, in conjunction with the dropped tokens optimization from Section 4.1, lead to smoothed ViTs having inference times comparable to standard (non-robust) models. For example, when using stride $s = 10$ and dropping masked tokens, a smoothed ViT-S is only 2x slower than a single inference step of a standard ResNet-50, while a smoothed ViT-B is only 5x slower. We report the inference time of these models, along with their standard and certified accuracies, in Table 1.

5 Related work

Certified defenses. An extensive body of research has studied the development of certified or provable defenses to adversarial perturbations. This line of research largely falls into one of three categories: tighter or exact verifiers [Kat+17; Ehl17; LM17; TXT19; Xia+19], convex relaxation-based defenses [WK18; RSL18; Won+18; Gow+18; Gow+19; MGV18; Wen+18; Zha+18; Sal+19a], and smoothing-based defenses [Lec+19; Li+18; CRK19; Sal+19b; LF20c; LF20a; Yan+20; Sal+20]. In the case of patches, the earliest certified defense used an instance of convex relaxation (interval bounds) to derive provable guarantees to adversarial patch [Chi+20]. Subsequent work [LF20b] focused on randomized smoothing. This approach smooths classifiers over random noise, but tend to be extremely expensive to use (4-5 orders of magnitudes slower than a standard, non-robust model) [CRK19; LF20b]. Recently, Lin et al. [Lin+21] proposed a variant based on randomized cropping that performs similarly to Levine and Feizi [LF20a] but with better guarantees under worse-case patch transformations.

Deterministic smoothing. To mitigate the expensive inference times of randomized smoothing, Levine and Feizi [LF20a] proposed derandomized smoothing, which used a finite set of ablations to smooth a base classifier. This substantially reduced the computational requirements of smoothing, but is still two orders of magnitude slower than standard models. Two similar defenses, Clipped BagNet [Zha+20] and PatchGuard [Xia+21], rely on restricting the model's receptive field. These approaches are faster than derandomized smoothing, but have other limitations. Clipped BagNet has substantially weaker robustness guarantees than derandomized smoothing. PatchGuard has higher but *brittle* guarantees: a defended model is optimally defended against a specific patch size, and achieves no robustness at all against patches that are even slightly larger than the one considered.

Empirical methods: attacks and defenses. Another line of work studies empirical approaches for generating adversarial patches and designing empirical defenses. Adversarial patches have been developed for downstream tasks such as image classification [KZG18], object detection [Eyk+18; Che+18; Liu+18], and facial recognition [Sha+16; TVRG19; BA18]. Several of these attacks work in the physical domain [Bro+18; Eyk+18; Che+18], and can successfully target tasks such as traffic sign recognition [Eyk+18; Che+18]. Heuristic defenses to these attacks include watermarking [Hay18] and gradient smoothing [NKP19]; however, these defenses were shown to be vulnerable adaptive attacks [Chi+20]. More recently, Rao, Stutz, and Schiele [RSS20] proposed an adversarial training approach to improve empirical robustness to patch attacks.

Vision transformers. Our work leverages the vision transformer (ViT) architecture [Dos+21], which adapts the popular attention-based model from the language setting [Vas+17] to the vision setting. Recent work [Tou+20] has released more efficient training methods as well as pre-trained ViTs that have made these architectures more accessible to the wider research community.

6 Conclusion

We demonstrate how applying visual transformers (ViTs) within the smoothing framework leads to significantly improved certified robustness to adversarial patches while maintaining standard accuracies that are on par with regular (non-robust) models. Further, we put forth changes to the ViT architecture and the corresponding smoothing procedure that greatly speed up the resulting inference times over previous smoothing approaches by up to two orders of magnitude—they end up being only 2-5x slower than that of a regular ResNet. We believe that these improvements finally establish models that are certifiably robust to adversarial patches as a viable alternative to standard (non-robust) models.

7 Acknowledgements

Work supported in part by the NSF grants CCF-1553428 and CNS-1815221, and Open Philanthropy. This material is based upon work supported by the Defense Advanced Research Projects Agency (DARPA) under Contract No. HR001120C0015.

Research was sponsored by the United States Air Force Research Laboratory and the United States Air Force Artificial Intelligence Accelerator and was accomplished under Cooperative Agreement Number FA8750-19-2-1000. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the United States Air Force or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein.

References

[BA18] Avishek Joey Bose and Parham Aarabi. "Adversarial attacks on face detectors using neural net based constrained optimization". In: *2018 IEEE 20th International Workshop on Multimedia Signal Processing (MMSP)*. 2018.

[BMV18] Mitali Bafna, Jack Murtagh, and Nikhil Vyas. "Thwarting Adversarial Examples: An *L*_0-RobustSparse Fourier Transform". In: *arXiv preprint arXiv:1812.05013* (2018).

[Bro+18] Tom B. Brown et al. *Adversarial Patch*. 2018. arXiv: 1712.09665 [cs.CV].

[Che+18] Shang-Tse Chen et al. "Shapeshifter: Robust physical adversarial attack on faster r-cnn object detector". In: *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer. 2018, pp. 52–68.

[Chi+20] Ping-yeh Chiang et al. "Certified defenses for adversarial patches". In: *arXiv preprint arXiv:2003.06693* (2020).

[CRK19] Jeremy M Cohen, Elan Rosenfeld, and J Zico Kolter. "Certified adversarial robustness via randomized smoothing". In: *International Conference on Machine Learning (ICML)*. 2019.

[Den+09] Jia Deng et al. "Imagenet: A large-scale hierarchical image database". In: *Computer Vision and Pattern Recognition (CVPR)*. 2009.

[Dos+21] Alexey Dosovitskiy et al. "An image is worth 16x16 words: Transformers for image recognition at scale". In: *International Conference on Learning Representations (ICLR)*. 2021.

[Ehl17] Rüdiger Ehlers. "Formal Verification of Piece-Wise Linear Feed-Forward Neural Networks". In: *Automated Technology for Verification and Analysis*. 2017.

[Evt+18] Ivan Evtimov et al. "Robust Physical-World Attacks on Machine Learning Models". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2018.

[Eyk+18] Kevin Eykholt et al. "Physical Adversarial Examples for Object Detectors". In: *CoRR* (2018).

[Gow+18] Sven Gowal et al. "On the Effectiveness of Interval Bound Propagation for Training Verifiably Robust Models". In: 2018.

[Gow+19] Sven Gowal et al. "Scalable verified training for provably robust image classification". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2019.

[Hay18] Jamie Hayes. "On visible adversarial perturbations & digital watermarking". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*. 2018, pp. 1597–1604.

[He+16] Kaiming He et al. "Deep Residual Learning for Image Recognition". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016.

[Hoo+20] Shahar Hoory et al. "Dynamic Adversarial Patch for Evading Object Detection Models". In: *arXiv preprint arXiv:2010.13070* (2020).

[JK19] Kyle D Julian and Mykel J Kochenderfer. "Guaranteeing safety for neural network-based aircraft collision avoidance systems". In: *2019 IEEE/AIAA 38th Digital Avionics Systems Conference (DASC)*. IEEE. 2019, pp. 1–10.

[Kat+17] Guy Katz et al. "Reluplex: An Efficient SMT Solver for Verifying Deep Neural Networks". In: *International Conference on Computer Aided Verification*. 2017.

[Kri09] Alex Krizhevsky. "Learning Multiple Layers of Features from Tiny Images". In: *Technical report*. 2009.

[KZG18] Danny Karmon, Daniel Zoran, and Yoav Goldberg. "Lavan: Localized and visible adversarial noise". In: *International Conference on Machine Learning*. PMLR. 2018, pp. 2507–2515.

[Lec+19] Mathias Lecuyer et al. "Certified robustness to adversarial examples with differential privacy". In: *Symposium on Security and Privacy (SP)*. 2019.

[LF20a] Alexander Levine and Soheil Feizi. "(De) Randomized Smoothing for Certifiable Defense against Patch Attacks". In: *arXiv preprint arXiv:2002.10733* (2020).

[LF20b] Alexander Levine and Soheil Feizi. "Robustness certificates for sparse adversarial attacks by randomized ablation". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 34. 04. 2020, pp. 4585–4593.

[LF20c] Alexander Levine and Soheil Feizi. "Wasserstein smoothing: Certified robustness against wasserstein adversarial attacks". In: *International Conference on Artificial Intelligence and Statistics*. PMLR. 2020, pp. 3938–3947.

[Li+18] Bai Li et al. "Certified adversarial robustness with additive noise". In: *arXiv preprint arXiv:1809.03113* (2018).

[Lin+21] Wan-Yi Lin et al. "Certified robustness against adversarial patch attacks via randomized cropping". In: *ICML 2021 Workshop on Adversarial Machine Learning*. 2021. URL: https://openreview.net/forum?id=g4NNK4RH715.

[Liu+18] Xin Liu et al. "Dpatch: An adversarial patch attack on object detectors". In: *arXiv preprint arXiv:1806.02299* (2018).

[LK19] Mark Lee and Zico Kolter. "On physical adversarial patches for object detection". In: *arXiv preprint arXiv:1906.11897* (2019).

[LM17] Alessio Lomuscio and Lalit Maganti. "An approach to reachability analysis for feed-forward ReLU neural networks". In: *ArXiv preprint arXiv:1706.07351*. 2017.

[MGV18] Matthew Mirman, Timon Gehr, and Martin Vechev. "Differentiable abstract interpretation for provably robust neural networks". In: *International Conference on Machine Learning (ICML)*. 2018.

[NKP19] Muzammal Naseer, Salman Khan, and Fatih Porikli. "Local gradients smoothing: Defense against localized adversarial attacks". In: *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE. 2019, pp. 1300–1307.

[Ran+19] Anurag Ranjan et al. "Attacking optical flow". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2019, pp. 2404–2413.

[RSL18] Aditi Raghunathan, Jacob Steinhardt, and Percy Liang. "Certified defenses against adversarial examples". In: *International Conference on Learning Representations (ICLR)*. 2018.

[RSS20] Sukrut Rao, David Stutz, and Bernt Schiele. "Adversarial training against location-optimized adversarial patches". In: *arXiv preprint arXiv:2005.02313* (2020).

[Rus+15] Olga Russakovsky et al. "ImageNet Large Scale Visual Recognition Challenge". In: *International Journal of Computer Vision (IJCV)*. 2015.

[Sal+19a] Hadi Salman et al. "A convex relaxation barrier to tight robustness verification of neural networks". In: *Advances in Neural Information Processing Systems (NeurIPS)* (2019).

[Sal+19b] Hadi Salman et al. "Provably robust deep learning via adversarially trained smoothed classifiers". In: *Advances in Neural Information Processing Systems (NeurIPS)*. 2019.

[Sal+20] Hadi Salman et al. "Denoised smoothing: A provable defense for pretrained classifiers". In: *Advances in Neural Information Processing Systems* 33 (2020).

[Sha+16] Mahmood Sharif et al. "Accessorize to a Crime: Real and Stealthy Attacks on State-of-the-Art Face Recognition". In: *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, Vienna, Austria, October 24-28, 2016*. 2016, pp. 1528–1540.

[Tou+20] Hugo Touvron et al. "Training data-efficient image transformers & distillation through attention". In: *arXiv preprint arXiv:2012.12877* (2020).

[Tra+20] Florian Tramer et al. "On adaptive attacks to adversarial example defenses". In: *arXiv preprint arXiv:2002.08347* (2020).

[TVRG19] Simen Thys, Wiebe Van Ranst, and Toon Goedemé. "Fooling automated surveillance cameras: adversarial patches to attack person detection". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*. 2019.

[TXT19] Vincent Tjeng, Kai Xiao, and Russ Tedrake. "Evaluating Robustness of Neural Networks with Mixed Integer Programming". In: *International Conference on Learning Representations (ICLR)*. 2019.

[Vas+17] Ashish Vaswani et al. "Attention is All you Need". In: *Advances in Neural Information Processing Systems* (2017).

[Wen+18] Tsui-Wei Weng et al. "Towards fast computation of certified robustness for ReLU networks". In: *International Conference on Machine Learning (ICML)*. 2018.

[Wig19] Ross Wightman. *PyTorch Image Models*. https://github.com/rwightman/pytorch-image-models. 2019. DOI: 10.5281/zenodo.4414861.

[WK18] Eric Wong and J Zico Kolter. "Provable defenses against adversarial examples via the convex outer adversarial polytope". In: *International Conference on Machine Learning (ICML)*. 2018.

[Won+18] Eric Wong et al. "Scaling provable adversarial defenses". In: *Advances in Neural Information Processing Systems (NeurIPS)*. 2018.

[Won+20] Eric Wong et al. "Neural network virtual sensors for fuel injection quantities with provable performance specifications". In: *2020 IEEE Intelligent Vehicles Symposium (IV)*. IEEE. 2020, pp. 1753–1758.

[Wu+20] Zuxuan Wu et al. "Making an invisibility cloak: Real world adversarial attacks on object detectors". In: *European Conference on Computer Vision*. Springer. 2020, pp. 1–17.

[Xia+19] Kai Y. Xiao et al. "Training for Faster Adversarial Robustness Verification via Inducing ReLU Stability". In: *International Conference on Learning Representations (ICLR)*. 2019.

[Xia+21] Chong Xiang et al. "PatchGuard: A Provably Robust Defense against Adversarial Patches via Small Receptive Fields and Masking". In: *30th {USENIX} Security Symposium ({USENIX} Security 21)*. 2021.

[Yan+20] Greg Yang et al. *Randomized Smoothing of All Shapes and Sizes*. 2020.

[Zha+18] Huan Zhang et al. "Efficient neural network robustness certification with general activation functions". In: *arXiv preprint arXiv:1811.00866* (2018).

[Zha+20] Zhanyuan Zhang et al. "Clipped BagNet: defending against sticker attacks with clipped bag-of-features". In: *2020 IEEE Security and Privacy Workshops (SPW)*. 2020.

[ZK16] Sergey Zagoruyko and Nikos Komodakis. "Wide residual networks". In: *arXiv preprint arXiv:1605.07146* (2016).

A Experimental setup

A.1 Models and architectures

We use three sizes of vision transformers—ViT-Tiny (ViT-T), ViT-Small (ViT-S), and ViT-Base (ViT-B) models [Wig19; Dos+21] and compare to to residual networks of similar (or larger) size—ResNet-18, ResNet-50 [He+16], and Wide ResNet-101-2 [ZK16], respectively. These architectures and their corresponding number of parameters are summarized in Table 4.

Table 4: A collection of neural network architectures we use in our paper.

Architecture	ViT-T	ResNet-18	ViT-S	ResNet-50	ViT-B	WRN-101-2
Params	5M	12M	22M	26M	86M	126M

We use the same architectures for both ImageNet and CIFAR-10 models, and finetune our smoothed models from publicly released checkpoints pretrained on ImageNet. All our CIFAR-10 experiments are thus conducted on upsampled CIFAR-10 images of size 224×224.

A.2 Datasets

We use two datasets:

1. CIFAR [Kri09] https://paperswithcode.com/dataset/cifar-10.
2. ImageNet [Rus+15], with a custom (research, non-commercial) license, as found here https://paperswithcode.com/dataset/imagenet.

A.3 Training parameters

Derandomized smoothing requires that the base classifier predict well on image ablations. A standard technique for derandomized smoothing methods is to directly train the base classifier on image ablations [LF20a]. Thus, unless otherwise stated, in each epoch we randomly apply a column ablation of fixed width to each image of the training set.

To facilitate training of the base classifiers, we start from pretrained ResNets[3] and ViT architectures[4] and fine-tune as follows:

ImageNet. We train for 30 epochs using SGD of fixed learning rate of 10^{-3}, a batch size of 256, a weight-decay of 10^{-4}, a momentum of 0.9, and with column ablations of fixed width $b = 19$. For data-augmentation, we use random resized crop, random horizontal flip, and color jitter. We then apply column ablations.

CIFAR-10. We train for 30 epochs using SGD with a step learning rate of 10^{-2} that drops every 10 epochs by a factor of 10, a batch size of 128, a weight-decay of 5×10^{-4}, a momentum of 0.9, and with column ablations of fixed width $b = 4$. We only use random horizontal flip for data-augmentation, after which we apply column ablations. We then upsample all CIFAR-10 images to 224×224 (on GPU).

Training time. Training is relatively fast, with our largest ImageNet model (WRN-101-2) finishing in roughly two days on one NVIDIA V100 GPU. The smaller models such as ViT-T or ResNet-18 finish training in only a few hours.

A.4 Compute and timing experiments

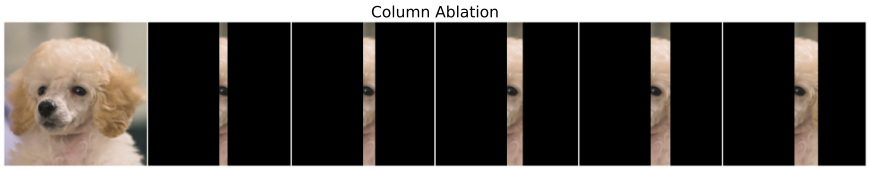
We use an internal cluster containing NVIDIA 1080-TI, 2080-TI, V100, and A100 GPUs. Scalability and timing experiments were performed on an A100 and averaged over 50 trials. When performing scalability experiments, we do not include data loading time or the time to move the input to the GPU.

A.5 Example ablations

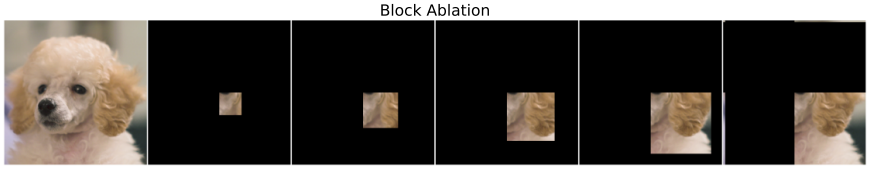
In Figure 7, we display examples of ablations of various types (column, block) and sizes.

[3] These are TorchVision's official checkpoints, and can be found here https://pytorch.org/vision/stable/models.html.

[4] We use the DeiT checkpoints of [Wig19] which can be found here https://github.com/rwightman/pytorch-image-models/blob/master/timm/models/vision_transformer.py.



(a) **Column ablations** with the following ablation size from left to right: original image, 13px, 19px, 25px, 31px, 37px.



(b) **Block ablations** with the following ablation size from left to right: original image, 35px, 55px, 75px, 95px, 115px.

Figure 7: Example ablations that we use in our paper.

A.6 Differences in setup from Levine and Feizi [LF20a]

Our work builds on top of that of Levine and Feizi [LF20a]. We use their robustness guarantee as is (see Section 2.1), but there are a few differences in the setup of our experiments. All experimental results (including the de-randomized smoothing baseline) are run using the same experimental setup in order to remain fair, which only improved the baseline over what was previously reported in the literature. For completeness, we describe the differences in setup here.

Encoding *null* inputs. The first difference is that Levine and Feizi [LF20a] encode part of the input as being *null* or ablated by adding additional color channels, as described in [LF20b], so that the *null* value is distinct from all real pixel colors. In practice, we found this to be unnecessary, and were able to replicate their results with ablations that simply replace masked pixels with 0.

Early stopping. We find that ResNets substantially benefit from early stopping when trained with ablations, and otherwise experience severe overfitting to the ablations with substantially reduced test accuracies. In our replication, we find that the ResNet-50 result reported by Levine and Feizi [LF20a] can be substantially improved with an earlier checkpoint (improving certified accuracy by nearly 10%), and thus we use early-stopping in all of our ResNet baselines.

Starting from pretrained models. To reduce training time, for both ImageNet and CIFAR-10 experiments, we start from pre-trained ImageNet checkpoints (see Section A.3). This step is especially necessary for the CIFAR-10 experiments, as it is quite challenging to train a ViT from scratch on CIFAR-10 (these models tend to require a large amount of data).

Upsampled CIFAR-10. In order to use the pretrained ImageNet checkpoints when training our base classifiers for CIFAR-10, we (nearest neighbor) upsample the CIFAR-10 inputs to 224×224 as part of the model architecture. We verify robustness in the original 32×32 images.

Sweeping over ablation size. We note that Levine and Feizi [LF20a] tested various ablations sizes only on CIFAR-10. Due to our speed-ups, we were able to sweep over ablations sizes for ImageNet.

B Ablation sweeps

In this section, we further explore the impact of changing the ablation size on both standard and certified performance. In Section B.1, we explore the effect of modifying the ablation size at training time. In Section B.2, similar to the experiment on ImageNet from Section 3.2, we present additional results on adjusting the ablation size at test time for CIFAR10.

B.1 Train-time ablation

We first explore varying the ablation size used during training for ImageNet. Specifically, we train and certify a ResNet-50 and ViT-S over a range of column widths from 1 to 67 pixels (Figure 8).

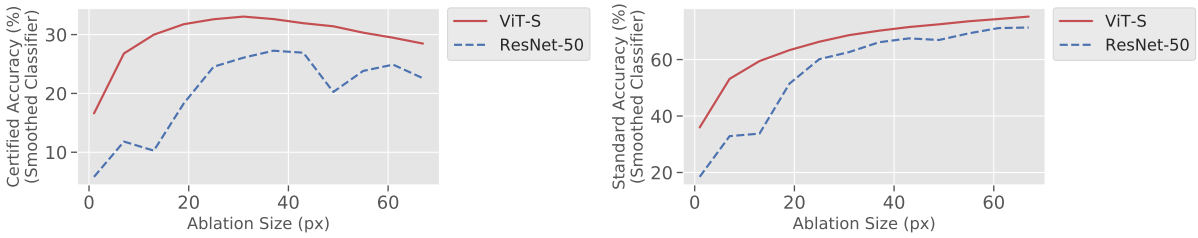


Figure 8: Certified and standard accuracy for a smoothed model trained and evaluated on ImageNet column ablations with varying widths. The ResNet-50 requires a substantially larger ablation size for certification, whereas the ViT-S is more flexible.

For ViTs, we find that once the columns are wide enough, we see only marginal improvements in certified accuracy (i.e. only 1.3% higher certified accuracy over $b = 19$). This suggests that small ablations are sufficient at training time, allowing for fast training of ViTs when using cropped ablations.

On the other hand, ResNets require a substantially larger column width than was previously explored. Specifically, the certified accuracy of the ResNet baseline can be greatly improved from 18% to 27% when the ablation size is increased to $b = 37$. This ablation size is optimal for the ResNet, but is still 6% lower certified accuracy when compared to the ViT.

Overall, we find that certified performance of ViTs on ImageNet remains largely stable with respect to the column ablation size used for training. We can thus use smaller ablation sizes during training (e.g $b = 19$) to improve training speed while certifying using larger ablation sizes.

B.2 Test-time ablations

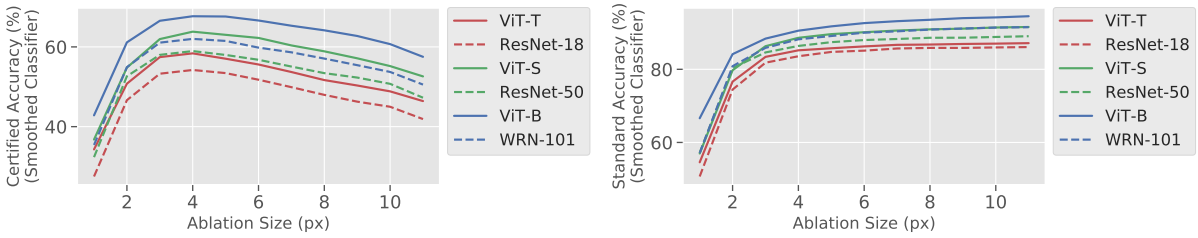


Figure 9: Certified and standard accuracy for a smoothed model on CIFAR-10 trained with a fixed ablation size ($b = 5$), and evaluated with varying ablation sizes.

Similar to the experiment on ImageNet from Section 3.2, we present analogous results for varying the ablation size used at test time for CIFAR-10. These results largely reflect what was previously observed by Levine and Feizi [LF20a]. Specifically, the optimal ablation size for CIFAR10 is a column width of $b = 4$, with a steep drop-off in performance for larger ablation sizes. This is in contrast to what we observed in ImageNet, which did not see such a steep drop in performance.

C Dropping tokens for ViTs

We first describe the algorithm for processing image ablations with a ViT while dropping masked tokens. Let $\mathbf{x}$ be an image with size $h \times w$, and let $\mathcal{S}(\mathbf{x})$ be the set of image ablations of $\mathbf{x}$. For each $\mathbf{z}, \mathbf{m} \in \mathcal{S}(\mathbf{x})$, $\mathbf{z}$ is an image ablation of size $h \times w$ and $\mathbf{m} \in \{0,1\}^{h \times w}$ is its corresponding mask, such that $\mathbf{m}_{ij}$ is 0 if the i, j pixel in $\mathbf{z}$ is masked and 1 otherwise.

Recall that a ViT has two stages when processing an input $\mathbf{z}$.

- **Encoding:** $\mathbf{z}$ is split into patches of $p \times p$ and positionally encoded into tokens. We let $E(\mathbf{w}, i, j)$ be an encoder which positionally encodes the $p \times p$ sized patch $\mathbf{w}$ which was at spatial location ip, jp in $\mathbf{z}$.
- **Self-Attention:** A set of positionally encoded tokens $\mathcal{T}$ is passed through self attention layers V and produces a class label.

Given an image ablation $\mathbf{z}$ we modify the ViT to remove tokens in $\mathcal{T}$ that correspond to a fully masked region in $\mathbf{z}$.

Algorithm 1 Forward pass for processing an image ablation $\mathbf{z}$ with mask $\mathbf{m}$ using a ViT while dropping masked tokens.

function PROCESSABLATION($\mathbf{z}, \mathbf{m}$)
  $\mathcal{T} = \{\}$ *Initialize set of tokens for an ablation*
  for $i, j \in [h/p] \times [w/p]$ **do**
    if not $\mathbf{m}_{ip:(i+1)p, jp:(j+1)p} = \mathbf{0}$ **then**
      $\mathcal{T} = \mathcal{T} \cup E(\mathbf{z}_{ip:(i+1)p, jp:(j+1)p}, i, j)$
    end if
  end for
  return $V(\mathcal{T})$
end function

We can then use this function to define the smoothed ViT.

Algorithm 2 Forward pass for a smoothed ViT on an input image $\mathbf{x}$ with ablation set $\mathcal{S}(\mathbf{x})$

function SMOOTHEDVIT($\mathbf{x}$)
  $c_i = 0$ for $i \in [k]$ *// Initialize counts to zero*
  for $\mathbf{z}, \mathbf{m} \in \mathcal{S}(\mathbf{x})$ **do**
    y = PROCESSABLATION($\mathbf{z}, \mathbf{m}$)
    $c_y = c_y + 1$ *// Update counts*
  end for
  return $\arg\max_y c_y$
end function

C.1 Computational complexity of ViTs with dropped tokens

We can now derive the computational complexity of the smoothed ViT when dropping tokens. Specifically, consider a ViT that divides an $h \times w$ pixel image into $p \times p$ patches, and positionally encodes them tokens with d hidden dimensions.

Recall that a ViT has two operation types: *attention operators* which scale quadratically with the number of tokens but linearly with hidden dimension d and *fully-connected operators* which scale linearly with the number of tokens but quadratically in d. Without dropping tokens, we have hw/p^2 tokens. A forward pass of processing an image ablation without dropping tokens thus has an overall complexity of

$$O\left(\left(\frac{hw}{p^2}\right)^2 d + \left(\frac{hw}{p^2}\right) d^2\right)$$

where the first term corresponds to the attention operations, and the second term corresponds to the fully-connected operations.

For column ablations with width b, dropping masked tokens reduces the number of tokens to hb/p^2. The complexity of the forward pass to process an image ablation when dropping masked tokens (i.e `ProcessAblation`) then drops

to

$$O\left(\left(\frac{hb}{p^2}\right)^2 d + \left(\frac{hb}{p^2}\right) d^2\right)$$

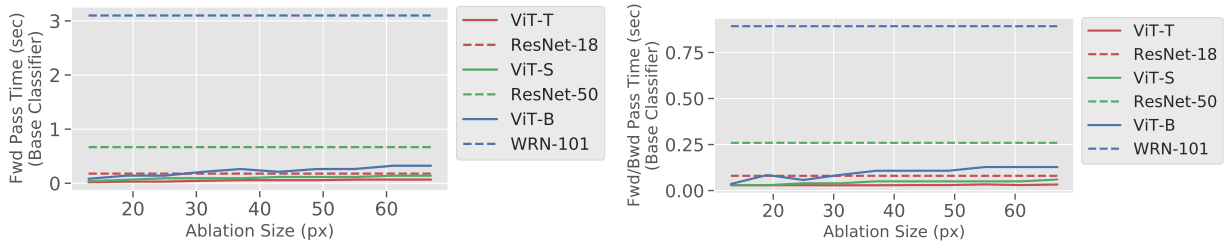
thus reducing the attention cost by a factor of $O(w^2/b^2)$ and the fully-connected cost by a factor of $O(w/b)$. In practice, the computation of fully-connected operations tends to dominate since $d > \frac{hw}{p^2}$.

Overall, a smoothed ViT with stride s processes w/s ablations. Thus, the overall complexity of the smoothed ViT is:

$$O\left(\frac{w}{s}\left(\left(\frac{hb}{p^2}\right)^2 d + \left(\frac{hb}{p^2}\right) d^2\right)\right)$$

C.2 Effect of dropping tokens on speed

We extend the timing experiments comparing ViTs and ResNets to a range of ablation sizes (previously presented in Table 3 from Section 4 for a single column ablation size of $b = 19$). Empirically, even for substantially larger ablations, we find significantly faster training and inference times for ViTs over ResNets. In Figure 10, we compare the evaluation and training speeds for processing image ablations with ResNets and ViTs with dropped tokens.



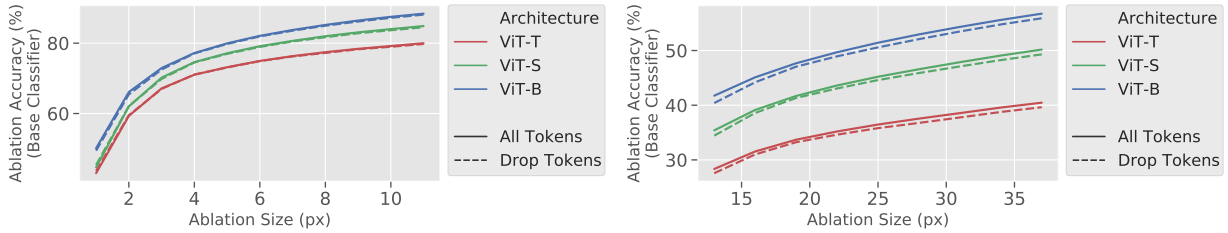
(a) Forward Pass Time

(b) Forward/Backward Pass Time

Figure 10: (a) Average time for computing a forward pass on a batch of 1024 image ablations on ImageNet (b) Average time for computing a full training step (forward and backward pass) on a batch of 128 image ablations on ImageNet

C.3 Effect of dropping tokens on performance

Since the tokens are individually positionally encoded, dropping tokens that are fully masked does not remove any information from the input. In Figure 11, we confirm that dropping masked tokens does not significantly change the accuracy of the ViT base classifier on ablations.



(a) CIFAR-10 models.

(b) ImageNet models.

Figure 11: We compare the ablation accuracies of dropping masked tokens versus processing all tokens for a collection of vision transformers on CIFAR-10 and ImageNet. Dropping masked tokens does not substantially degrade accuracy.

D Strided ablations

In this section, we explore strided ablations for certification in more depth. In Section D.1 we present the threshold for certification when using strided ablations. In Section D.2 we show how striding affects performance.

D.1 Certification thresholds for strided ablation sets

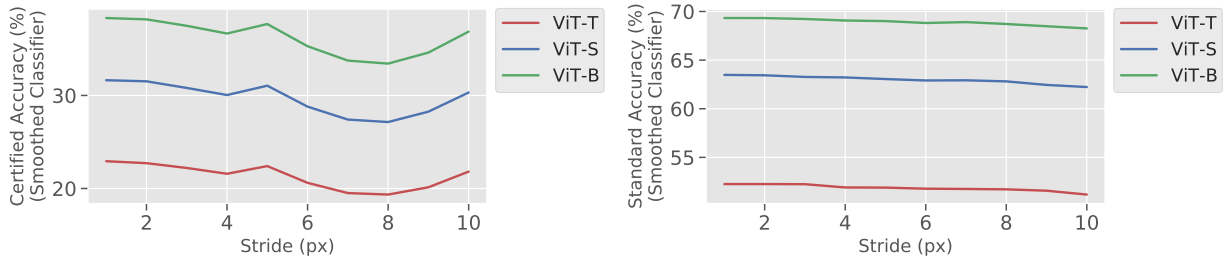
We briefly describe the new thresholds for certification when using strided ablations. Recall from equation 2 that a prediction is certified robust if

$$n_c(\mathbf{x}) > \max_{c' \neq c} n_{c'}(\mathbf{x}) + 2\Delta.$$

Thus Δ, the number of ablations that a patch can intersect, fully describes the certification threshold.

Column smoothing. For column smoothing with width b and stride s, the maximum number of ablations that an $m \times m$ patch can intersect with is at most $\Delta_{column+stride} = \lceil (m + s - 1)/s \rceil$.

D.2 Performance under strided ablations



(a) Certified Accuracy

(b) Standard Accuracy

Figure 12: Certified and standard accuracy of various ViTs for ImageNet when using strided column ablations with varying stride lengths.

In this section, we explore how striding affects standard and certified performance. We find that striding does not result in a monotonic change in certified accuracy—certification accuracy can both decrease and increase as the stride increases.

For a few choices in striding, it is possible to not substantially change the accuracy of the ViT at classifying ablations, as shown in Figure 12. For example, a ViT-B which normally obtains 38.3% certified accuracy without striding, maintains certified accuracies of 37.6% at stride $s = 5$ and 36.8% at stride $s = 10$. For these small drops in certified accuracy, striding directly enables 5x or 10x faster inference times.

E Block smoothing

In this section, we investigate an alternative type of smoothing known as *Block Smoothing*, previously investigated in the CIFAR-10 setting [LF20a]. In block smoothing, we ablate (square) blocks of pixels instead of columns of pixels. This procedure is prohibitively expensive for ImageNet due to its quadratic complexity. For example, smoothing a 224×224 image with block ablations takes a majority vote over $224 \times 224 = 50,176$ ablations, which is four orders of magnitude slower than a standard forward pass. We alleviate this obstacle for larger image settings such as ImageNet with the token-based speedups for ViTs from Section 4.1 and the striding from Section 4.2. In combination, these improvements in speed allow us to perform a practical investigation into block smoothing on ImageNet.

Certification. Certification of derandomized smoothing models with block ablations is similar to that of column ablations, and depends on the maximum number of ablations in the ablation set that an adversarial patch can simultaneously intersect. Recall that for column ablations of size b, the certification threshold is $\Delta = m + b - 1$ ablations. For block ablations of size b (where b here is the side of the retained block/square of pixels), $\Delta = (m + b - 1)^2$. The threshold can then be plugged as before into Equation equation 2 to check whether the model is certifiably robust.

E.1 Practical inference speeds for block smoothing

We first demonstrate how dropping masked tokens significantly increases the speed of evaluating block ablations for the base classifier. In Figure 13, we show that dropping masked tokens substantially reduces the time needed to process 1024 block ablations for various sizes of ViTs. This directly leads to a 4.85x speedup for ViT-S with ablation size 75.

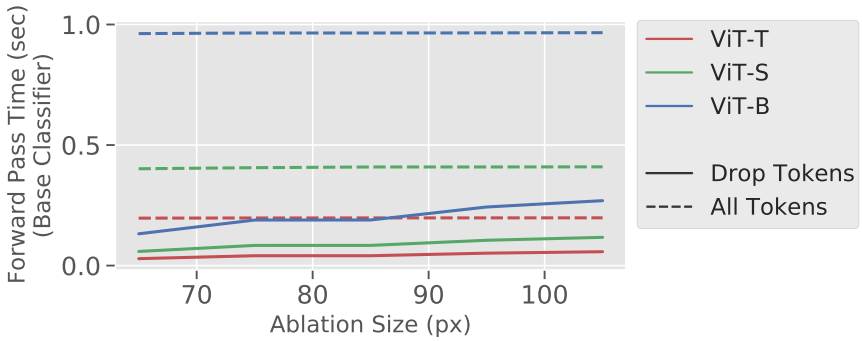


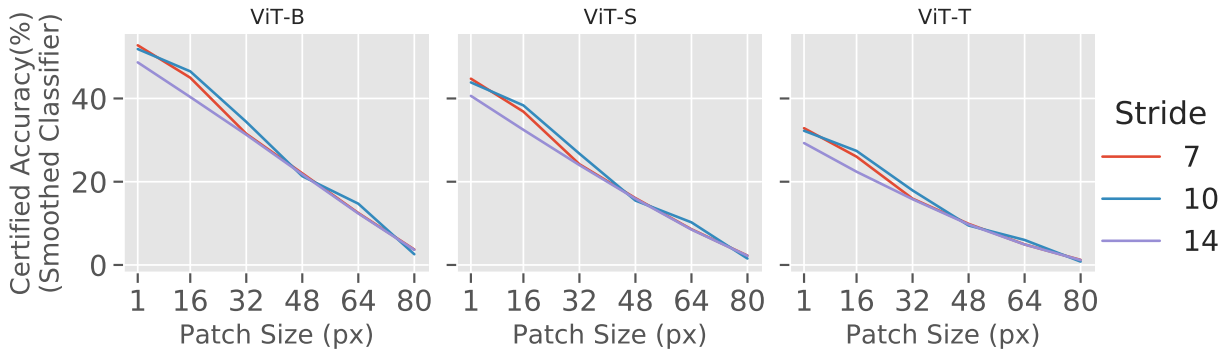
Figure 13: Average time to compute a forward pass for ViTs on 1024 block ablated images with varying ablation sizes with and without dropping masked tokens.

Even with this optimization, however, block smoothing is quite expensive. A forward pass through the smoothed model still requires around 50k passes through the base classifier. We thus leverage our second speedup from strided ablations and use *strided* block smoothing. Similar to strided column ablations, for a stride length of s, we only consider block ablations that are s pixels apart, vertically and horizontally. This changes the certification threshold Δ to be, $\Delta_{block+stride} = \lceil (m + s - 1)/s \rceil^2$. With dropping fully masked tokens and using a stride of 10, a smoothed ViT-S using an ablation size of 75 is only 2.8x slower than a standard (non-robust) ResNet-50.

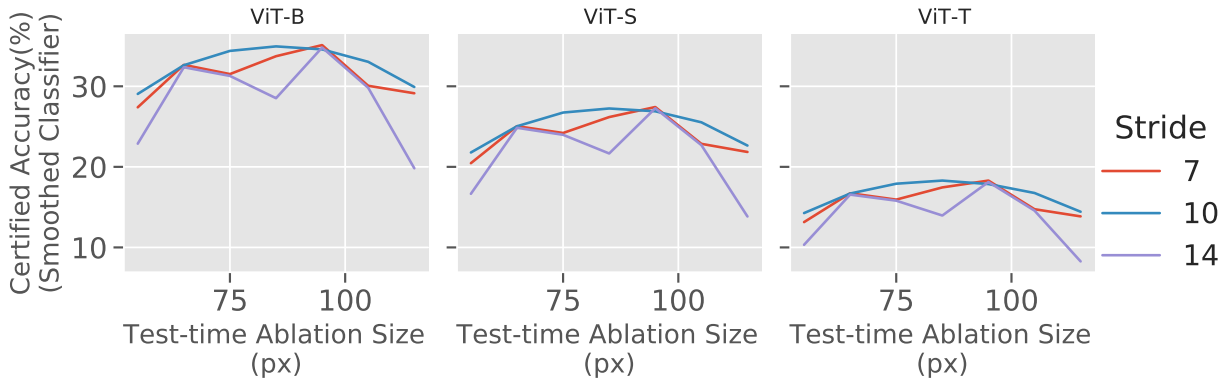
Certified accuracy. We find that, despite an systematic search over stride length and block size (both at training and evaluation), block smoothing on ImageNet remains significantly worse than column smoothing. For example, with optimal stride and ablation size, we see up to 5% lower certified accuracy than column smoothing on the largest model, ViT-B. We checked a range of ablation sizes from 55 to 115 as well as three stride lengths $\{7, 10, 14\}$ (Figure 14).

Similar to striding with column ablations, there is a significant amount of variability based on the stride length. To pinpoint the effect of striding, we certify one of the best-performing block sizes ($b = 75$) over a full range of strides from $s = 1$ to $s = 20$ (Figure 15). This is a fairly expensive calculation, as using stride $s = 1$ corresponds to the full block ablation with 50k ablations.

Even when using all possible block ablations ($s = 1$), block smoothing does not improve over column smoothing. However, we do find that certain stride lengths ($s = 18$) can achieve similar performance to non-strided block ablations ($s = 1$), which means that we can speed up certification (by 18x) without sacrificing certified accuracy. Thus, while our methods can make block smoothing computationally feasible, further investigation is needed to make block smoothing match column smoothing in terms of certified and standard accuracies.



(a) We fix the test-time ablation size at $b = 75$ and plot the certified accuracy as a function of the adversarial patch size, for various stride length.



(b) We fix the adversarial patch size $m = 32$ and plot the certified accuracy as a function of the test-time ablation size, for various stride length.

Figure 14: Strided block smoothing on ImageNet for a collection of ViT models trained with block ablations of size $b = 75$.

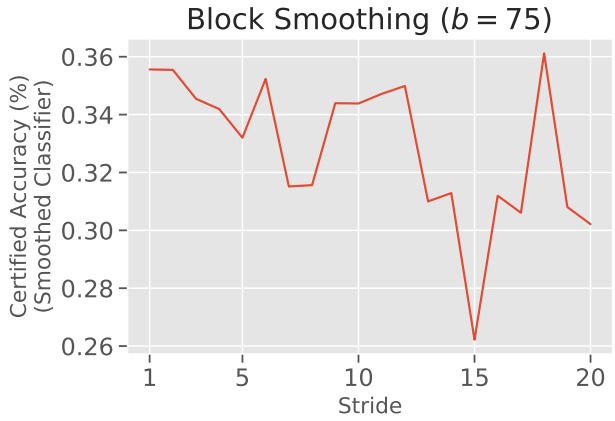


Figure 15: Strided block smoothing on ImageNet for ViT-B with a fixed ablation size $b = 75$. The reported certified accuracy are against adversarial patches of size 32×32. Note how some stride lengths ($s = 18$ for example) can achieve similar performance to non-strided block ablations ($s = 1$).

F Extended experimental results

Table 5: **An extended version of Table 1.** Summary of our ImageNet results and comparisons to certified patch defenses from the literature: Clipped Bagnet (CBG), Derandomized Smoothing (DS), and PatchGuard (PG). Time refers to the inference time for a batch of 1024 images, b is the ablation size, and s is the ablation stride.

Standard and Certified Accuracy on ImageNet (%)					
Patch Size	Clean	1% pixels	2% pixels	3% pixels	Time (sec)
CBN [Zha+20]	49.5	13.4	7.1	3.1	3.05 ± 0.02
DS [LF20a]	44.4	17.7	14.0	11.2	149.52 ± 0.33
PG [Xia+21] (1% pixels)	55.1	32.3	0.0	0.0	3.05 ± 0.02
PG [Xia+21] (2% pixels)	54.6	26.0	26.0	0.0	3.05 ± 0.02
PG [Xia+21] (3% pixels)	54.1	19.7	19.7	19.7	3.05 ± 0.02
Vary Ablation Size (Stride = 1)					
ResNet-18 (b = 19)	50.6	24.1	19.8	16.9	39.84 ± 0.97
ResNet-18 (b = 25)	52.7	24.2	20.0	17.1	39.84 ± 0.97
ResNet-18 (b = 37)	54.3	22.4	18.6	15.7	39.84 ± 0.97
ViT-T (b = 19)	52.3	27.3	22.9	19.9	6.81 ± 0.05
ViT-T (b = 25)	53.7	26.9	22.8	19.7	6.82 ± 0.05
ViT-T (b = 37)	55.6	25.5	21.7	18.8	12.64 ± 0.10
ResNet-50 (b = 19)	51.5	22.8	18.3	15.3	149.52 ± 0.33
ResNet-50 (b = 25)	54.7	23.8	19.5	16.4	149.52 ± 0.33
ResNet-50 (b = 37)	57.8	23.1	19.0	16.1	149.52 ± 0.33
ViT-S (b = 19)	63.5	36.8	31.6	27.9	14.00 ± 0.16
ViT-S (b = 25)	65.1	36.8	31.9	28.2	20.58 ± 0.18
ViT-S (b = 37)	67.1	35.3	30.7	27.1	20.61 ± 0.16
WRN-101-2 (b = 19)	61.4	33.3	28.1	24.1	694.50 ± 0.58
WRN-101-2 (b = 25)	64.2	34.3	29.1	25.3	694.50 ± 0.58
WRN-101-2 (b = 37)	67.2	33.7	28.8	25.2	694.50 ± 0.58
ViT-B (b = 19)	69.3	43.8	38.3	34.3	31.51 ± 0.17
ViT-B (b = 25)	71.1	44.0	38.8	34.8	31.52 ± 0.21
ViT-B (b = 37)	73.2	43.0	38.2	34.1	58.74 ± 0.17
Vary Ablation Stride					
WRN-101-2 (b = 19, s = 5)	61.1	30.1	27.3	21.9	138.90 ± 0.12
WRN-101-2 (b = 19, s = 10)	59.7	25.8	25.8	20.9	69.45 ± 0.06
ViT-B (b = 19, s = 5)	69.0	40.6	37.7	32.0	6.30 ± 0.03
ViT-B (b = 19, s = 10)	68.3	36.9	36.9	31.4	3.15 ± 0.02
WRN-101-2 (b = 37, s = 5)	66.9	32.6	27.2	24.7	138.90 ± 0.12
WRN-101-2 (b = 37, s = 10)	66.1	31.7	26.7	21.7	69.45 ± 0.06
ViT-B (b = 37, s = 5)	73.1	41.9	36.4	33.5	11.75 ± 0.03
ViT-B (b = 37, s = 10)	72.6	41.3	36.1	30.8	5.87 ± 0.02

Table 6: **An extended version of Table 2**. Summary of our CIFAR-10 results and comparisons to certified patch defenses from the literature: Clipped Bagnet (CBG), Derandomized Smoothing (DS), and PatchGuard (PG). b is the column ablation size out of 32 pixels.

Standard and Certified Accuracy on CIFAR-10 (%)			
Patch Size	Clean	2×2	4×4
Baselines			
CBN [Zha+20]	84.2	44.2	9.3
DS [LF20a]	83.9	68.9	56.2
PG [Xia+21] (2×2)	84.7	69.2	0.0
PG [Xia+21] (4×4)	84.6	57.7	57.7
Smoothed models			
ResNet-18 (b = 4)	83.6	67.0	54.2
ViT-T (b = 4)	**85.5**	**70.0**	**58.5**
ResNet-50 (b = 4)	86.4	71.6	59.0
ViT-S (b = 4)	**88.4**	**75.0**	**63.8**
WRN-101-2 (b = 4)	88.2	73.9	62.0
ViT-B (b = 4)	**90.8**	**78.1**	**67.6**

Table 7: Standard accuracies of regularly trained architectures vs. smoothed architectures with column ablations of size $b = 4$ for CIFAR-10 and $b = 19$ for ImageNet.

		Standard accuracy of architecture (%)					
		ViT-T	ResNet-18	ViT-S	ResNet-50	ViT-B	WRN-101-2
ImageNet	Standard	72.03	69.76	79.72	76.13	81.74	78.85
	Smoothed	52.25	50.62	63.48	51.47	69.33	61.38
	Difference	19.77	19.14	16.24	24.66	12.41	17.47
CIFAR-10	Standard	93.13	95.72	93.33	96.16	97.07	97.85
	Smoothed	85.53	88.41	86.39	83.57	90.75	88.20
	Difference	7.60	7.31	6.94	12.59	6.32	9.65

Table 8: ImageNet certified models trained with ablations of size 19, with a variety of test-time ablations sizes b and ablation stride lengths s.

Architecture	s	b	Standard accuracy(%)	Certified Accuracy(%) 1% pixels	2% pixels	3% pixels
ResNet-18	1	19	50.6	24.1	19.8	16.9
		25	52.7	24.2	20.0	17.1
		37	54.3	22.4	18.6	15.7
	5	19	50.3	21.5	19.3	15.3
		25	52.4	22.1	17.9	15.8
		37	54.2	21.5	17.4	15.4
	10	19	49.3	18.7	18.7	14.8
		25	51.5	21.5	17.3	13.6
		37	53.7	21.0	17.1	13.5
ViT-T	1	19	52.3	**27.3**	**22.9**	**19.9**
		25	53.7	26.9	22.8	19.7
		37	**55.6**	25.5	21.7	18.8
	5	19	51.9	24.6	22.4	18.2
		25	53.5	25.1	20.6	18.5
		37	55.4	24.7	20.5	18.5
	10	19	51.2	21.8	21.8	17.8
		25	53.1	24.6	20.4	16.4
		37	55.1	24.4	20.3	16.5
ResNet-50	1	19	51.5	22.8	18.3	15.3
		25	54.7	23.8	19.5	16.4
		37	57.8	23.1	19.0	16.1
	5	19	51.0	20.1	17.9	13.6
		25	54.5	21.7	17.2	15.1
		37	57.7	22.1	17.7	15.8
	10	19	49.9	17.2	17.2	13.2
		25	53.7	21.0	16.7	12.8
		37	57.1	21.7	17.6	13.7
ViT-S	1	19	63.5	**36.8**	31.6	27.9
		25	65.1	**36.8**	**31.9**	**28.2**
		37	**67.1**	35.3	30.7	27.1
	5	19	63.1	33.8	31.1	25.7
		25	64.9	34.4	29.3	26.7
		37	67.0	34.3	29.1	26.7
	10	19	62.2	30.3	30.3	25.2
		25	64.3	33.9	28.7	23.7
		37	66.5	33.8	29.0	24.2
WRN-101	1	19	61.4	33.3	28.1	24.1
		25	64.2	34.3	29.1	25.3
		37	67.2	33.7	28.8	25.2
	5	19	61.1	30.1	27.3	21.9
		25	63.8	31.8	26.3	23.7
		37	66.9	32.6	27.2	24.7
	10	19	59.7	25.8	25.8	20.9
		25	62.7	30.5	25.3	20.5
		37	66.1	31.7	26.7	21.7
ViT-B	1	19	69.3	43.8	38.3	34.3
		25	71.1	**44.0**	**38.8**	**34.8**
		37	**73.2**	43.0	38.2	34.1
	5	19	69.0	40.6	37.7	32.0
		25	70.8	41.6	36.0	33.0
		37	73.1	41.9	36.4	33.5
	10	19	68.3	36.9	36.9	31.4
		25	70.3	40.9	35.2	29.8
		37	72.6	41.3	36.1	30.8