On the Interplay of Convolutional Padding and Adversarial Robustness

Paul Gavrikov[1*] Janis Keuper[1,2*]
[1]IMLA, Offenburg University, [2]Fraunhofer ITWM
{paul.gavrikov,janis.keuper}@hs-offenburg.de

Abstract

It is common practice to apply padding prior to convolution operations to preserve the resolution of feature-maps in Convolutional Neural Networks (CNN). While many alternatives exist, this is often achieved by adding a border of zeros around the inputs. In this work, we show that adversarial attacks often result in perturbation anomalies at the image boundaries, which are the areas where padding is used. Consequently, we aim to provide an analysis of the interplay between padding and adversarial attacks and seek an answer to the question of how different padding modes (or their absence) affect adversarial robustness in various scenarios.

1. Introduction

Over the recent years, Convolutional Neural Networks (CNN) [1] have become the dominant backbone of most learning-based approaches for computer vision applications [2]. However, despite their overwhelming success in terms of achieving high test accuracies on various (vision) benchmarks, CNNs also have shown to be very vulnerable against minor changes in the input data distribution, *e.g.*, against adversarial attacks [3]. This is especially concerning in safety-critical applications such as autonomous driving [4] or medical imaging [5] where human lives are at stake. Beyond adversarial training [6], which can be considered as the current standard approach towards the training of more robust networks, recent works showed that inherent signal processing flaws related to convolutions are one of several possible sources for the lack of robustness. While prior investigations, including down-sampling [7, 8], the size of the convolution kernels [9, 10] and the choice of the nonlinear activation functions [11, 12], have shown a significant impact of these network components on the robustness of trained CNN models, there is no systematic analysis of the commonly applied padding schemes in this context.

Our contribution is motivated by an initial analysis of spatial

*Funded by the German Ministry for Science, Research and Arts, Baden-Wuerttemberg under Grant 32-7545.20/45/1 (Q-AMeLiA).

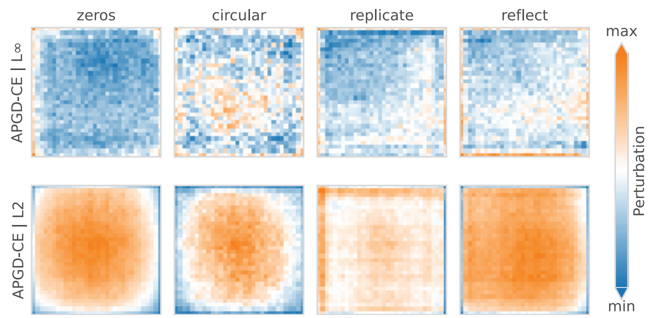


Figure 1: Average perturbations of 1,000 *CIFAR-10* samples where attacks were successful. Adversarial attacks show perturbation anomalies (more or fewer perturbations) at image boundaries that can be attributed to padding. Shown here is a *ResNet-20* trained without adversarial defenses with different padding modes under attacks of high-budget ℓ_∞- (top row) and ℓ_2-bounded (bottom row) *APGD-CE*.

attack locations (see fig. 1), which showed strong anomalies of attack intensities at the boundaries of attacked images. Hence, we investigate the role of padding and its parameters (like type and size) for the robustness of CNNs.

The key contributions of this paper are:

- We provide the first in-depth analysis of the impact of the padding-related architectural design choices in the context of adversarial CNN robustness.
- Our empirical evaluations on *CIFAR-10* [13] show that the commonly applied *same* sized *zero* padding does not always result in the best performance, especially in combination with adversarial training. However, standard benchmarks such as *AutoAttack* [14] fail to reflect this.
- Additionally, we also investigate padding-free architectures, *i.e.*, by using up-scaling or out-painting to increase the input image/feature-map size to compensate for the down-scaling effect of non-padded convolutions.

zeros reflect replicate circular

Figure 2: Examples of a *CIFAR-10* training sample increased to an image resolution to 64 × 64 px by different padding approaches.

2. Related Work

Adversarial robustness. Neural networks tend to overfit the training data distribution and fail to generalize beyond. As such the predictions are often highly sensitive to small input perturbations [15, 16] that are (almost) imperceptible and semantically meaningless to humans. In some cases, these perturbations can be as small as a single pixel [17]. Formally, this phenomenon can be introduced as follows. Given a model $\mathcal{F}$ parameterized by θ, an input sample x with the corresponding class label y, and a loss function $\mathcal{L}$, an adversarial attack will attempt to maximize the loss $\mathcal{L}$ by finding an additive perturbation to x in the $\mathcal{B}_\epsilon(x)$ ball that is centered at x. The ℓ_p-norm denoted by $\|\cdot\|_p$ is bounded by the radius (budget) ϵ to restrict perturbations to minor changes.

$$\begin{gathered}\max_{x' \in \mathcal{B}_\epsilon(x)} \mathcal{L}\left(\mathcal{F}\left(x';\theta\right), y\right) \\ \mathcal{B}_\epsilon(x) = \{x' : \|x - x'\|_p \leq \epsilon\}\end{gathered} \tag{1}$$

Adversarial attacks can be found in both, white- and black-box settings [18, 19, 20, 21]. Amongst the most effective attacks are gradient-based white-box attacks that use the model prediction to perturb images in the direction of the highest loss [22, 6, 14, 23]. Models trained without adversarial defenses can typically not withstand attacks with high ϵ budgets. An unequivocal solution to overcome this phenomenon is *adversarial training (AT)* [6]. Adversarial training trains the model on worst-case perturbations found during training and effectively turns out-of-domain attacks to in-domain samples. As a side-effect, this results in models that classify based on shapes and not texture information which is better aligned with human vision [24, 25]. Further, they are less over-confident [26] than normally trained ones. Unfortunately, adversarial training is also susceptible to overfitting to the attacks employed during its training phase [27, 28, 29]. Hence, a common choice to assess robustness is *AutoAttack* [14] which compares against multiple attacks [30] such as *APGD-CE* [14], *FAB* [23], and *Square* [21].

Correlation between architecture and robustness. *Tang et al.* [31] performed a mass evaluation of different networks against robustness. They concluded that architecture plays an important role in robustness but no universal training recipe exists. For example, they recommend training light-weight architectures with AdamW [32], but SGD [33] performs significantly better for heavier architectures. Further, they state that CNNs outperform transformers on natural and system noise, but the opposite holds for adversarial robustness. *Huang et al.* [34] tested the influence of topology, kernel size, activation, normalization, and network size of residual networks on robustness. They find that under the same FLOPs, narrow-deep networks are more robust than shallow-wide networks and derive an optimal ratio. Further, they observe that pre-activation increases robustness, but increasing kernel size does not. Robust architectures can also be an optimization goal in Neural Architecture Search [35]. *Gavrikov et al.* [36, 37] provided an analysis in weight space and showed that convolution filters of adversarially trained CNNs are more diverse in learned patterns and generally occupy more of the network capacity than normally trained counterparts.

Padding. Prior to convolution, inputs can be artificially enlarged (padding), *e.g.*, to maintain the resolution before and after convolutions. Padding is controlled by two parameters: the *padding mode* which determines the information in the border, and the *padding size* which determines the size of the border. Popular frameworks like PyTorch provide a multitude of padding modes (see Fig. 2 for examples):

- **zeros:** the padding area is filled with zeros.
- **reflect:** mirrors the input at the boundary.
- **replicate:** copies the last pixel at the image boundary into the padding area.
- **circular:** wraps the input around the boundary and continues at the opposite boundary(s).

Despite the variety in choice, few recent works have studied the importance of padding in convolution layers, and the majority of image classification networks after *AlexNet* [38] use *zero* padding. In theory, the padding size can be set independently on all image axes, most commonly though, it is kept similar constant and determined by the kernel size k: $\lfloor k/2 \rfloor$ (also called *same* padding). For instance, for $k = 3$ the padding size for *same* padding is 1. Multiple works showed that *zero* padding increases translational invariance [39, 40, 41] and *circular* padding appears to break it [39]. However, the commonly used *same zero* padding can cause the model to produce artifacts in feature-maps that result in the loss of visual features (blind spots) [42]. The authors link this to even input resolutions and show that switching to uneven resolutions improves accuracy. Although all other padding modes except zero padding reduce artifacts in feature-maps, the authors cannot recommend a single best padding mode, as the best choice appears to depend on the specific problem [43, 44, 45].

To the best of our knowledge, no investigation of the role of padding on robustness has been performed so far. We aim to close this gap in this work.

3. Experiments

For our experiments, we train models and switch the padding mode in all convolution layers between *zeros*, *reflect*, *replicate*, and *circular*. Further, we switch the convolution kernel size $k \in \{3, 5, 7, 9\}$ and set the padding size to $\lfloor k/2 \rfloor$ per side, accordingly (*same* padding). The trained models are attacked with *APGD-CE* [14], *FAB* [23], and *Square* [21] using the implementation in [14]. The attacks on models are performed under two different budgets, derived from trends in recent literature. For *low budget* attacks we generate attacks from $p = 2, \epsilon = 0.1$ and $p = \infty, \epsilon = 1/255$; and for *high budget* attacks $p = 2, \epsilon = 0.5$ and $p = \infty, \epsilon = 8/255$. Attacks are evaluated on a subset of 1,000 *CIFAR-10* test samples. We measure the clean performance and robust performance under individual attacks and store and analyze perturbed inputs that have successfully fooled the model.

Training Details. We train 20-layer deep ResNets optimized for *CIFAR-10* [13] (*ResNet-20*) as introduced in [46] models on *CIFAR-10* [13] with the default train/validation splits. Training images are randomly horizontally flipped during training. Test images are not modified. Both splits are normalized by the channel mean and standard deviation. Training is executed with an SGD [33] optimizer (with Nesterov momentum [47] of 0.9) for 75 epochs, with an initial learning rate of 0.01 following a cosine annealing schedule [48], a weight decay [49] of 0.01, a batch size of 256, and cross-entropy loss [50] with a label smoothing [51] value of 0.1. For our analyses, we use the model checkpoints at the end of training. For adversarial training experiments, we follow the same parameters except that we train with an FGSM [22] $p = \infty, \epsilon = 8/255$ adversary and select the checkpoint with the highest accuracy against a PGD [52] $p = \infty, \epsilon = 8/255$ adversary (early stopping) [22] on the test data to prevent *robust overfitting* [53, 27, 28]. Although neither our training nor tested architecture is comparable with SOTA approaches (*e.g.*, [54, 55, 56, 57, 34]), we believe it to be sufficiently well suited for this analysis. Unless stated otherwise we report results over 10 model runs with different randomness seeds.

3.1. Clean Performance

On clean data and without adversarial training, we see that *zero* padding outperforms all other padding modes in clean accuracy (Tab. 1). However, under adversarial training *replicate* or *reflect* outperform *zero* padding, depending on the kernel size. While the difference is not very high with larger kernels, at the common kernel size $k = 3$ it amounts to 1.27%. By a large margin, *circular* padding performs the worst in combination with adversarial training at all kernel sizes. We also observe that the accuracy falls off with increasing kernel size without adversarial training, while there is hardly a correlation on adversarially-trained models. However, it is worth noting that $k > 3$ models appear to reach higher accuracy under adversarial training.

Table 1: Clean test accuracy of *ResNet-20* trained on *CIFAR-10* with different padding modes, kernel sizes, and adversarial training. Mean over 10 runs. **Best**, <u>second best</u>. We do not report the second best option for all normally trained runs with $k < 9$ as the differences between non-*zeros* padding modes are insignificant.

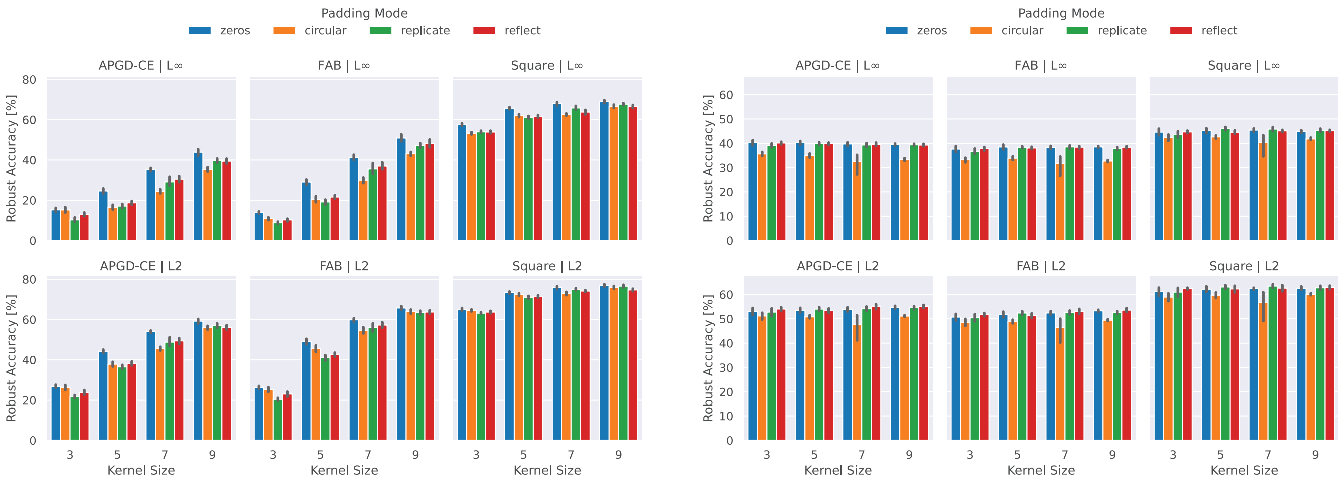
		Clean Test Performance [%] (↑)			
AT	k	zeros	circular	replicate	reflect
✗	3	**90.26**	90.10	90.13	90.15
	5	**90.14**	89.66	89.82	89.67
	7	**89.36**	88.49	88.52	88.47
	9	**88.22**	<u>87.50</u>	87.03	87.25
✓	3	<u>71.84</u>	69.17	70.79	**73.11**
	5	<u>73.72</u>	71.34	**74.02**	73.08
	7	<u>73.86</u>	67.33	**73.89**	73.10
	9	<u>73.51</u>	71.53	72.24	**73.90**

Table 2: *AutoAttack* [14] robust test accuracy of *ResNet-20* trained on *CIFAR-10* with different padding modes, kernel sizes, and adversarial training. Mean over 10 runs. **Best**, <u>second best</u>.

		AutoAttack Performance [%] (↑)			
AT	k	zeros	circular	replicate	reflect
✗	3	**8.52**	4.69	4.90	<u>5.79</u>
	5	**17.69**	10.44	11.12	<u>12.33</u>
	7	**29.06**	17.86	<u>24.55</u>	24.35
	9	**39.18**	30.52	<u>36.39</u>	34.81
✓	3	**36.88**	32.09	35.91	<u>36.82</u>
	5	**37.48**	32.34	<u>37.30</u>	37.12
	7	**37.42**	30.16	37.08	<u>37.26</u>
	9	**37.49**	31.09	36.89	<u>37.25</u>

3.2. Robust Performance

Next, we compare the robust accuracy under different attacks while testing different padding approaches and kernel sizes (Fig. 3).



(a) Native robustness (low budget).

(b) Robustness after adversarial training (high budget).

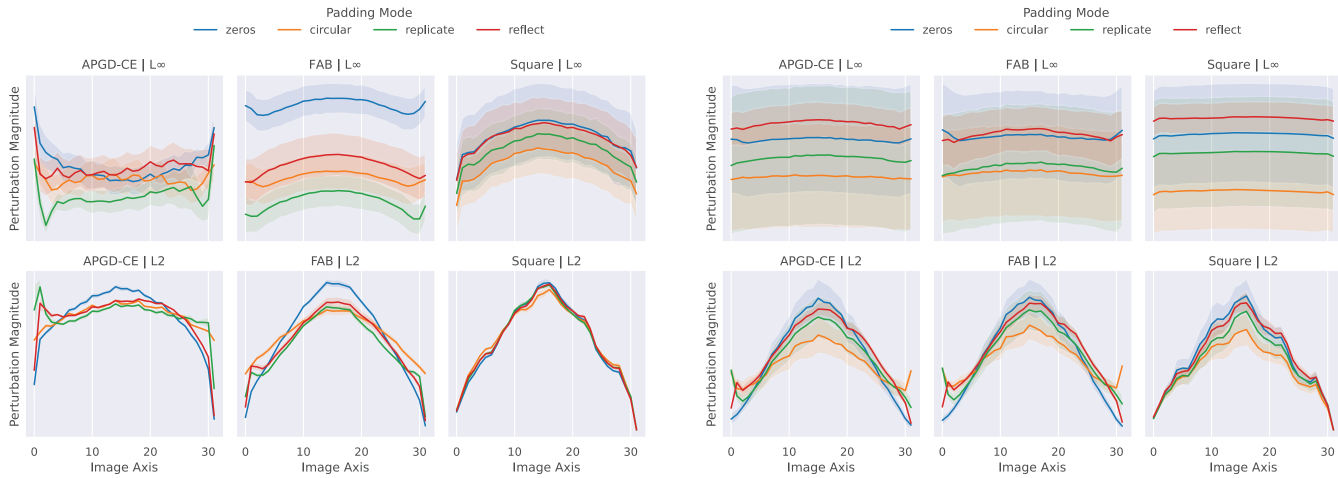
Figure 3: Robust accuracy under different attacks on a *ResNet-20* trained on *CIFAR-10* with different padding modes and kernel sizes. Fig. 3a low-budget attacks on normally trained models. Fig. 3b high-budget attacks on adversarially trained models. Variance computed over 10 models.

Native Robustness. First, we analyze the performance of models trained without adversarial defenses such as adversarial training. For low budgets, we observe that *zeros* outperforms all other padding modes in most settings by a large margin (Fig. 3a). Generally, we see a very similar trend for padding modes independent of the individual attack or norm, of course, except for the actual robust performance (*FAB* is the strongest attack, and *Square* is the weakest). Most notably, we observe that increasing k seems to diminish the gap between *zero* padding and other modes. Interestingly, contrary to the observations of [34] we find that increasing kernel size also significantly improves robustness in this setting. For example, we see an improvement by 28.64% between $k = 3$ and $k = 9$ kernels against ℓ_∞-bounded *APGD-CE* adversaries. As expected, when switching to high-budget attacks (not pictured for brevity), all tested padding mode/kernel size combinations collapse below random performance and mostly even to near 0 performance. For $k = 3$, *circular* seems to be the only interesting outlier, as it gains a few percent in robust accuracy, yet, still underperforms a random baseline.

Adversarial Training. Additionally to native robustness, we aim to understand whether differences arise in models trained with adversarial training (Fig. 3b). Contrary to previous results, observing patterns in this setting becomes less straightforward. Averaging over all attacks, we again see that robustness improves with increasing kernel size, albeit at an almost insignificant rate for $k \geq 5$. Additionally, all padding modes seem to perform reasonably well but depending on the attack, norm, or kernel size individual modes perform better than others. The only exception to this is *circular* padding. Although it performed reasonably well regarding native robustness, in combination with adversarial training it is always by far the worst choice. Regarding the other padding modes, we make the following observations: If we average the robust performance over all attacks and norms we see the same patterns as for clean accuracy, *reflect* is the best choice for $k = 3, 9$, *replicate* performs best for $k = 5, 7$. However, we also see large fluctuations. For example, for $k = 5$ *reflect* becomes the worst choice after *circular* with *replicate* being the best, on $k = 7$ the difference between *reflect* and *replicate* is marginal as both perform almost equally well and, finally, at $k = 9$ we see a switch and *replicate* becomes the worst choice after *circular* while *reflect* becomes the best choice. Interestingly, ***zero* padding is never the best option on average over all attacks**. Yet, if we only analyze ℓ_∞-bounded *APGD-CE* attacks, it outperforms all other padding modes by a non-negligible margin. This is a concern when evaluating with *AutoAttack* [14] which starts with *APGD-CE* attacks, and proceeds with other attacks only on samples that could not be successfully attacked. As such it is not surprising that in terms of *AutoAttack* accuracy, *zero* padding remains the best-performing method Tab. 2. However, *replicate/reflect* padding modes are usually only marginally worse and come at the benefit of improved clean accuracy.

3.3. Padding Anomalies

Following our robustness performance analysis, we now aim to understand where perturbations are primarily located and how different padding modes influence their distribu-



(a) Normal training (high budget attack).

(b) Adversarial training (high budget attack).

Figure 4: Distribution of perturbation magnitudes of successful attacks along the image X-axis under different padding modes, and different attacks on $k = 3$ models. Fig. 4a shows high-budget attacks on normally trained models and Fig. 4b high-budget attacks on adversarially trained models. Variance computed over 10 models.

tion on $k = 3$ models. To this end, we average the absolute differences between successful adversarial examples (i. e. those that flip the predicted label) and the corresponding clean samples (Fig. 1). We separate between normally-trained (native) models and adversarially-trained models, but this time evaluate both under high budgets to increase perturbations aiming to improve visibility. For a better comparison between tested dimensions, we project perturbation magnitudes to the image X-axis (Fig. 4). Generally, though, we see relatively similar observations on the Y-axis, except that perturbations are less symmetrically distributed and increase towards the lower image edge.

Native Models. We see significant differences (Fig. 4a) between ℓ_2-perturbations which are primarily located in the center of the image and ℓ_∞-perturbations which appear to be distributed more uniformly across the image but contain anomalies in outermost pixels or their close neighbors. Noticeably, these are areas where the receptive field intersects with the padded area. For ℓ_∞-bounded *APGD-CE*, and to a lesser extent also *FAB*, we see increases in perturbations at boundaries, while *Square* shows strong decreases in perturbations at boundaries. Generally, we see similar distribution shapes independent of padding mode, except for ℓ_2-bounded *APGD-CE* where *reflect/replicate* show increased perturbations at the boundaries. However, we see a different area under the curve (AUC) for padding modes and *zero* shows the largest AUC.

Adversarial Training. Under adversarial training (Fig. 4b) perturbation distributions of ℓ_∞-bounded attackers level out but show a significantly increased variance between runs. Anomalies at the boundaries vanish for the largest part but are still slightly noticeable on all padding modes except *circular*. *Reflect* now has the largest AUC. For ℓ_2-bound adversaries we see relatively similar distributions, even in AUC, except for *circular* which has the lowest AUC overall. Boundary anomalies are again visible for *replicate/reflect*, but now also for *circular*.

3.4. The Computational Overhead of Padding

We have seen that *zero* padding may not always be the best choice, depending on the training type and the attack used. However, it has one significant advantage over the other padding modes used as it is entirely independent of the image content. Theoretically, this would mean that *zero* padding would be the computationally cheapest option. To understand if this also holds on real hardware that often optimizes processes we benchmark the required time for padding. In addition to just padding we also measure the time for a 2D convolution (32 filters with 3×3 kernel size and 1 px padding on all sides) operation on single input (32×32 resolution) for a more realistic scenario. Both operations are evaluated on GPU. We run the experiments on an NVIDIA A100-SXM4-40GB GPU with CUDA 11.3, cuDNN 8302, using PyTorch 1.12.1. All measurements are reported over 10,000 independent trials.

The results in Tab. 3 show an interesting trend. For the padding operation alone, *zeros* doubles the required time in comparison to *reflect* and *replicate*, but requires only a third of the time of *circular*. However, in terms of the total budget for the convolution *zeros* yields the fastest forward pass. *Reflect* and *replicate* are approximately 35% slower,

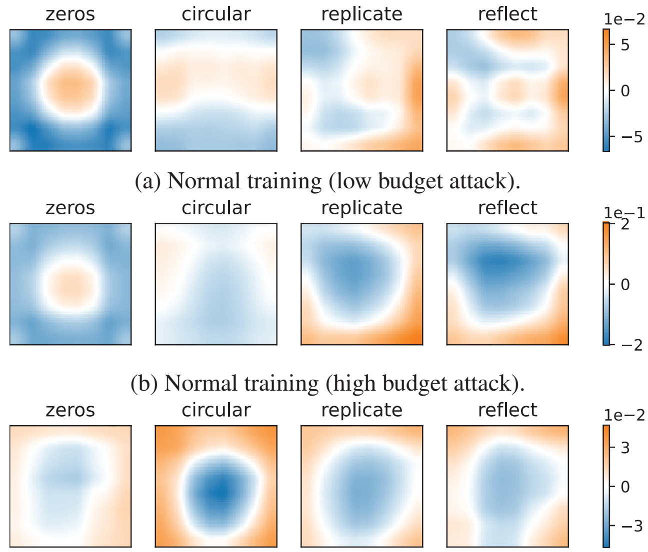
Table 3: Benchmark of the average time for padding or padded 2D convolution operation under different padding modes. Bold marks **best**.

Operation	Average time for operation [μs] ($\downarrow$) zeros	reflect	replicate	circular
Only padding	21.87	12.55	**10.10**	56.96
2D Convolution	**55.65**	76.86	74.04	132.60

and *circular* introduces an overhead of approximately 137% - presumably, due to optimized implementations. Our results show that the superiority of *replicate*, *reflect* in some settings comes at the cost of slower test and thus also train time.

3.5. Effect on Model Decisions

Our previous results showed that the choice of padding method and size can have significant effects on both, the model robustness (Tab. 1 and Fig. 3) and the placement of the perturbations (Fig. 1). To further analyze whether the perturbation anomalies at image edges are affecting the model decision intrinsics, and thus the robustness, we analyze visual explanations via *LayerCAM* [58] as implemented in [59]. Explanations are computed for the feature-maps after the last residual block with respect to the predicted label. We compute the differences in explanations between an adversarial example and its clean counterpart whenever the attack was successful and visualize the mean over all samples of 10 differently seeded models (Fig. 5). For this analysis, we limit ourselves to ℓ_∞-bounded high-budget *APGD-CE* attacks on $k = 3$ models. There is a clear shift in explanations between normally trained and adversarially-trained models. For normally trained models under low-budget attacks (Fig. 5a) we observe a difference in the explanation shift depending on the padding mode. With *zero* padding, explanations clearly shift toward the center indicating that the attack actually concentrates on the image foreground (as *CIFAR-10* is well-centered). Under *circular* padding the shift area seems to stretch horizontally and attacks start to aim at image boundaries. *Replicate* and *reflect* padding modes show almost similar behavior: while there is some shift towards the center, there are increasingly more attacks on the image boundary (except for the top left corner) than with previous modes. As we increase the attack budget (Fig. 5b) we observe the same behavior for all padding modes except *zero* padding. Attacks move entirely to the image boundaries padding showing that the adversary attacks everything **except** the image foreground. The shift intensity is smaller for *circular* than for *replicate/reflect*. For *zero* padding we observe hardly any differences to the low-budget setting. For adversarially-trained models (Fig. 5c) we see quite the opposite: attacks for all



(a) Normal training (low budget attack).

(b) Normal training (high budget attack).

(c) Adversarial training (high budget attack).

Figure 5: Average shifts in *LayerCAM* [58] explanations following successful ℓ_∞-bounded *APGD-CE* attacks under different padding modes. Mean computed over 10 models. Orange areas indicate areas where explanations move under adversarial attacks, while blue areas indicate areas of explanations have been removed.

padding modes are shifted toward the image edges, with no clear difference between padding modes except for the increased intensity of shifts in *circular* models. Although not pictured here for brevity, we observe similar trends for ℓ_2-bounded attacks.

3.6. Is No Padding a Better Alternative?

In this next section, we aim to understand the role of padding per se on robustness by entirely removing padding from all convolution layers, and without padding, models should not develop boundary anomalies. Without any other changes, this would result in smaller feature-map (representation) resolutions, and, residual networks would experience a size mismatch between input signals propagated between skip connections and the actual convolution paths. To solve the size mismatch after skip connections we zero-pad the processed signal before the summation instead of directly summing the residual and processed signal. Note that this is different from zero-padded convolutions as it does not result in zero-padded inputs to the next operators. Regarding the smaller feature-maps, we compare $k = 3$ padding-free networks in 3 settings:

- **Unmodified (None):** We use the same 32×32 px *CIFAR-10* samples as before. For *ResNet-20* this results in 4×4 representations at the deepest layers instead of 8×8.

Figure 6: Examples of *CIFAR-10* validation samples (top row) increased to an image resolution to 48×48 px by *MAT* [61] (bottom row).

- **Upscaling (None + Up):** We upscale *CIFAR-10* samples to 48×48 px via bilinear interpolation [60] to match the representations of padded networks.
- **Outpainting (None + Out):** To increase the resolution in a meaningful manner we experiment with image outpainting to 48×48 px via *MAT* [61] pre-trained on the *Places* dataset [62] containing scenic images that should not interfere with the original label (Fig. 6). We outpaint both, train and test samples.

Without adversarial training, we see a drop in clean and robust accuracy if we simply disable padding without countermeasures (Tab. 4). With both, upscaling or outpainting, we are able to mitigate the drop in clean accuracy, yet provoke a complete failure in robust accuracy except for *Square* attacks but still at a decreased robustness. Although it is worth noting that ℓ_∞-bounded attacks can attack more area for the same budget when increasing the image resolution as we do in upscaling or outpainting, the delta is too large to be caused by only an increased attack area. Under adversarial training, we see similar trends, but this time we also measure a clear difference between upscaling and outpainting. Surprisingly, outpainting seems even further to reduce robustness. However, we also notice that outpainting seems to affect non-masked areas (the original sample) and tampers with image statistics. In all cases, disabling padding results in worse performance and is therefore not recommended.

4. Conclusion & Discussion

We have evaluated the adversarial robustness of *CIFAR-10* models under different padding modes, kernel sizes, attacks, and training modes. Based on our results, we can provide the following recommendations:
For settings targeting native robustness, we advise using *zero* padding and increasing the kernel/padding size, if possible. We hypothesize that larger kernels are more likely to cause robustness than larger padding sizes. Switching to other padding modes clearly deteriorates the performance

Table 4: Performance comparison of padding-free $k = 3$ models against zero padding. Mean over 3 runs.

		Test Accuracy [%] (↑)						
			ℓ_2			ℓ_∞		
Padding	**AT**	**Clean**	APGD-CE	FAB	Square	APGD-CE	FAB	Square
Zeros	✗	**90.26**	**26.87**	**26.24**	**65.10**	**15.23**	**13.87**	**57.55**
None	✗	87.19	16.90	19.23	56.40	9.80	13.47	49.47
None + Up	✗	89.69	3.03	1.03	50.70	0.20	0.00	29.97
None + Out	✗	89.69	3.03	1.03	50.70	0.20	0.00	29.97
Zeros	✓	**71.84**	**52.93**	**50.72**	**61.17**	**40.22**	**37.62**	**44.59**
None	✓	67.40	48.80	46.73	56.87	38.60	35.93	41.50
None + Up	✓	47.80	40.33	38.47	45.60	31.50	28.00	31.03
None + Out	✓	40.62	32.63	30.07	38.43	29.00	24.80	26.27

and starts shifting attacks toward image boundaries.
When using adversarial training, it can be fruitful to experiment with the *reflect* and *replicate* padding modes. However, commonly models are benchmarked with *AutoAttack* [14] which starts with *APGD-CE* attacks where *zero* padding performs best, and proceeds with other attacks only on samples that could not be successfully attacked. Since *APGD-CE* is usually a very effective attack, the difference measured based on the remaining subset may not be significant to show clear differences between padding modes. However, at negligible impairments of robustness performance, we saw an improved clean accuracy of non-*zero* padding modes. In addition, we suggest reporting adversarial robustness for multiple attacks separately when studying padding.
Lastly, in all cases, we can recommend the usage of padding as padding-free architectures performed significantly worse in all investigated scenarios.

Limitations. For completeness, we want to emphasize that we only experimented on *CIFAR-10*. As with many "toy-datasets", objects in question are usually perfectly centered in the images. Unfortunately, this applies to most common benchmarking datasets such as *MNIST* [63], *SVHN* [64], *CIFAR-100* [13], *ImageNet* [65] etc. As such, we are not confident that we would obtain significantly different results on these datasets. Repeating our experiments on less curated datasets may, however, result in different observations. Further, we only experimented with ResNet-20. While we believe that the results are representative of the popular residual networks in general, we cannot guarantee that our results scale to entirely different architectures. We aim to bridge these gaps in future work.

References

[1] Yann LeCun, Yoshua Bengio, *et al.*, "Convolutional networks for images, speech, and time series," 1

[2] Ákos Zarándy, Csaba Rekeczky, Péter Szolgay, and Leon O Chua, "Overview of cnn research: 25 years history and the current trends," in *2015 IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 401–404, IEEE, 2015. 1

[3] Naveed Akhtar and Ajmal Mian, "Threat of adversarial attacks on deep learning in computer vision: A survey," *Ieee Access*, vol. 6, pp. 14410–14430, 2018. 1

[4] Yulong Cao, Chaowei Xiao, Benjamin Cyr, Yimeng Zhou, Won Park, Sara Rampazzi, Qi Alfred Chen, Kevin Fu, and Z. Morley Mao, "Adversarial sensor attack on lidar-based perception in autonomous driving," in *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, CCS '19, (New York, NY, USA), p. 2267–2281, Association for Computing Machinery, 2019. 1

[5] Samuel G. Finlayson, John D. Bowers, Joichi Ito, Jonathan L. Zittrain, Andrew L. Beam, and Isaac S. Kohane, "Adversarial attacks on medical machine learning," *Science*, vol. 363, no. 6433, pp. 1287–1289, 2019. 1

[6] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu, "Towards deep learning models resistant to adversarial attacks," in *International Conference on Learning Representations*, 2018. 1, 2

[7] Julia Grabinski, Janis Keuper, and Margret Keuper, "Aliasing and adversarial robust generalization of cnns," *Machine Learning*, 2022. 1

[8] Julia Grabinski, Steffen Jung, Janis Keuper, and Margret Keuper, "Frequencylowcut pooling - plug and play against catastrophic overfitting," in *Computer Vision - ECCV 2022 - 17th European Conference, Tel Aviv, Israel, October 23-27, 2022, Proceedings, Part XIV*, pp. 36–57, Springer, 2022. 1

[9] Ricard Durall, Margret Keuper, and Janis Keuper, "Watch your up-convolution: Cnn based generative deep neural networks are failing to reproduce spectral distributions," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 7890–7899, 2020. 1

[10] Nergis Tomen and Jan C van Gemert, "Spectral leakage and rethinking the kernel size in cnns," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 5138–5147, 2021. 1

[11] Kai Xiao, Vincent Tjeng, Nur Muhammad Shafiullah, and Aleksander Madry, "Training for faster adversarial robustness verification via inducing relu stability," in *International Conference on Learning Representations*, no. 2019, 2019. 1

[12] Lily Weng, Huan Zhang, Hongge Chen, Zhao Song, Cho-Jui Hsieh, Luca Daniel, Duane Boning, and Inderjit Dhillon, "Towards fast computation of certified robustness for relu networks," in *International Conference on Machine Learning*, pp. 5276–5285, PMLR, 2018. 1

[13] Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton, "Cifar-10 (canadian institute for advanced research)," 2009. 1, 3, 7

[14] Francesco Croce and Matthias Hein, "Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks," in *ICML*, 2020. 1, 2, 3, 4, 7

[15] Battista Biggio, Igino Corona, Davide Maiorca, Blaine Nelson, Nedim Šrndić, Pavel Laskov, Giorgio Giacinto, and Fabio Roli, "Evasion attacks against machine learning at test time," in *Machine Learning and Knowledge Discovery in Databases* (Hendrik Blockeel, Kristian Kersting, Siegfried Nijssen, and Filip Železný, eds.), (Berlin, Heidelberg), pp. 387–402, Springer Berlin Heidelberg, 2013. 2

[16] 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, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings* (Yoshua Bengio and Yann LeCun, eds.), 2014. 2

[17] Jiawei Su, Danilo Vasconcellos Vargas, and Kouichi Sakurai, "One pixel attack for fooling deep neural networks," *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 5, pp. 828–841, 2019. 2

[18] Yanpei Liu, Xinyun Chen, Chang Liu, and Dawn Song, "Delving into transferable adversarial examples and black-box attacks," in *International Conference on Learning Representations*, 2017. 2

[19] Andrew Ilyas, Logan Engstrom, Anish Athalye, and Jessy Lin, "Black-box adversarial attacks with limited queries and information," in *Proceedings of the 35th International Conference on Machine Learning, ICML 2018*, July 2018. 2

[20] Arjun Nitin Bhagoji, Warren He, Bo Li, and Dawn Song, "Practical Black-Box Attacks on Deep Neural Networks Using Efficient Query Mechanisms," in *Computer Vision – ECCV 2018*, pp. 158–174, Cham, Switzerland: Springer, Oct. 2018. 2

[21] Maksym Andriushchenko, Francesco Croce, Nicolas Flammarion, and Matthias Hein, "Square attack: A query-efficient black-box adversarial attack via random search," in *Computer Vision – ECCV 2020* (Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, eds.), (Cham), pp. 484–501, Springer International Publishing, 2020. 2, 3

[22] Ian Goodfellow, Jonathon Shlens, and Christian Szegedy, "Explaining and harnessing adversarial examples," in *International Conference on Learning Representations*, 2015. 2, 3

[23] Francesco Croce and Matthias Hein, "Minimally distorted adversarial examples with a fast adaptive boundary attack," 2020. 2, 3

[24] Paul Gavrikov, Janis Keuper, and Margret Keuper, "An extended study of human-like behavior under adversarial training," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pp. 2360–2367, June 2023. 2

[25] Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A. Wichmann, and Wieland Brendel, "Imagenet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness.," in *International Conference on Learning Representations*, 2019. 2

[26] Julia Grabinski, Paul Gavrikov, Janis Keuper, and Margret Keuper, "Robust models are less over-confident," in *Advances in Neural Information Processing Systems* (Alice H.

Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, eds.), 2022. 2

[27] Eric Wong, Leslie Rice, and J. Zico Kolter, "Fast is better than free: Revisiting adversarial training," in *International Conference on Learning Representations*, 2020. 2, 3

[28] Maksym Andriushchenko and Nicolas Flammarion, "Understanding and improving fast adversarial training," in *Advances in Neural Information Processing Systems* (H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, eds.), vol. 33, pp. 16048–16059, Curran Associates, Inc., 2020. 2, 3

[29] Hoki Kim, Woojin Lee, and Jaewook Lee, "Understanding catastrophic overfitting in single-step adversarial training," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, pp. 8119–8127, May 2021. 2

[30] Nicholas Carlini, Anish Athalye, Nicolas Papernot, Wieland Brendel, Jonas Rauber, Dimitris Tsipras, Ian Goodfellow, Aleksander Madry, and Alexey Kurakin, "On evaluating adversarial robustness," *arXiv preprint arXiv:1902.06705*, 2019. 2

[31] Shiyu Tang, Ruihao Gong, Yan Wang, Aishan Liu, Jiakai Wang, Xinyun Chen, Fengwei Yu, Xianglong Liu, Dawn Song, Alan Yuille, Philip H. S. Torr, and Dacheng Tao, "Robustart: Benchmarking robustness on architecture design and training techniques," 2021. 2

[32] Ilya Loshchilov and Frank Hutter, "Decoupled weight decay regularization," in *International Conference on Learning Representations*, 2019. 2

[33] Herbert E. Robbins, "A stochastic approximation method," *Annals of Mathematical Statistics*, vol. 22, pp. 400–407, 1951. 2, 3

[34] Shihua Huang, Zhichao Lu, Kalyanmoy Deb, and Vishnu Naresh Boddeti, "Revisiting residual networks for adversarial robustness: An architectural perspective," 2022. 2, 3, 4

[35] Steffen Jung, Jovita Lukasik, and Margret Keuper, "Neural architecture design and robustness: A dataset," in *The Eleventh International Conference on Learning Representations*, 2023. 2

[36] Paul Gavrikov and Janis Keuper, "CNN Filter DB: An empirical investigation of trained convolutional filters," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 19066–19076, June 2022. 2

[37] Paul Gavrikov and Janis Keuper, "Adversarial Robustness Through the Lens of Convolutional Filters," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pp. 139–147, June 2022. 2

[38] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems* (F. Pereira, C.J. Burges, L. Bottou, and K.Q. Weinberger, eds.), vol. 25, Curran Associates, Inc., 2012. 2

[39] Jonas Geiping, Hartmut Bauermeister, Hannah Dröge, and Michael Moeller, "Inverting gradients - how easy is it to break privacy in federated learning?," in *Advances in Neural Information Processing Systems* (H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, eds.), vol. 33, pp. 16937–16947, Curran Associates, Inc., 2020. 2

[40] Md Amirul Islam, Sen Jia, and Neil D. B. Bruce, "How much position information do convolutional neural networks encode?," in *International Conference on Learning Representations*, 2020. 2

[41] Osman Semih Kayhan and Jan C. van Gemert, "On translation invariance in cnns: Convolutional layers can exploit absolute spatial location," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 2

[42] Bilal Alsallakh, Narine Kokhlikyan, Vivek Miglani, Jun Yuan, and Orion Reblitz-Richardson, "Mind the pad – CNNs can develop blind spots," in *International Conference on Learning Representations*, 2021. 2

[43] Pedro O. Pinheiro, Tsung-Yi Lin, Ronan Collobert, and Piotr Dollár, "Learning to refine object segments," in *Computer Vision – ECCV 2016* (Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, eds.), (Cham), pp. 75–91, Springer International Publishing, 2016. 3

[44] Stefan Schubert, Peer Neubert, Johannes Pöschmann, and Peter Protzel, "Circular convolutional neural networks for panoramic images and laser data," *2019 IEEE Intelligent Vehicles Symposium (IV)*, pp. 653–660, 2019. 3

[45] Shikhar Vashishth, Soumya Sanyal, Vikram Nitin, Nilesh Agrawal, and Partha Talukdar, "Interacte: Improving convolution-based knowledge graph embeddings by increasing feature interactions," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, pp. 3009–3016, Apr. 2020. 3

[46] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," 2015. 3

[47] Yurii Nesterov, "A method for unconstrained convex minimization problem with the rate of convergence $o(1/k^2)$," *Doklady Akademii Nauk*, vol. 269, pp. 543–547, 1983. 3

[48] Ilya Loshchilov and Frank Hutter, "SGDR: Stochastic gradient descent with warm restarts," in *International Conference on Learning Representations*, 2017. 3

[49] Anders Krogh and John A. Hertz, "A simple weight decay can improve generalization," in *Proceedings of the 4th International Conference on Neural Information Processing Systems*, NIPS'91, (San Francisco, CA, USA), p. 950–957, Morgan Kaufmann Publishers Inc., 1991. 3

[50] Irving John Good, "Rational decisions," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 14, no. 1, pp. 107–114, 1952. 3

[51] Ian Goodfellow, Yoshua Bengio, and Aaron Courville, *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org. 3

[52] Alexey Kurakin, Ian J. Goodfellow, and Samy Bengio, "Adversarial machine learning at scale," in *International Conference on Learning Representations*, 2017. 3

[53] Leslie Rice, Eric Wong, and J. Zico Kolter, "Overfitting in adversarially robust deep learning," in *Proceedings of the 37th International Conference on Machine Learning*, ICML'20, JMLR.org, 2020. 3

[54] Qiyu Kang, Yang Song, Qinxu Ding, and Wee Peng Tay, "Stable neural ODE with lyapunov-stable equilibrium points for defending against adversarial attacks," in *Advances in Neural Information Processing Systems* (A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, eds.), 2021. 3

[55] Sylvestre-Alvise Rebuffi, Sven Gowal, Dan Andrei Calian, Florian Stimberg, Olivia Wiles, and Timothy Mann, "Data augmentation can improve robustness," in *Advances in Neural Information Processing Systems* (A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, eds.), 2021. 3

[56] Sven Gowal, Sylvestre-Alvise Rebuffi, Olivia Wiles, Florian Stimberg, Dan Andrei Calian, and Timothy Mann, "Improving robustness using generated data," in *Advances in Neural Information Processing Systems* (A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, eds.), 2021. 3

[57] Sven Gowal, Chongli Qin, Jonathan Uesato, Timothy Mann, and Pushmeet Kohli, "Uncovering the limits of adversarial training against norm-bounded adversarial examples," 2020. 3

[58] Peng-Tao Jiang, Chang-Bin Zhang, Qibin Hou, Ming-Ming Cheng, and Yunchao Wei, "Layercam: Exploring hierarchical class activation maps for localization," *IEEE Transactions on Image Processing*, 2021. 6

[59] Jacob Gildenblat and contributors, "Pytorch library for cam methods." `https://github.com/jacobgil/pytorch-grad-cam`, 2021. 6

[60] Hsieh Hou and H. Andrews, "Cubic splines for image interpolation and digital filtering," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 26, no. 6, pp. 508–517, 1978. 7

[61] Wenbo Li, Zhe Lin, Kun Zhou, Lu Qi, Yi Wang, and Jiaya Jia, "MAT: Mask-aware transformer for large hole image inpainting," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022. 7

[62] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba, "Places: A 10 million image database for scene recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 6, pp. 1452–1464, 2018. 7

[63] Yann LeCun, Corinna Cortes, and CJ Burges, "Mnist handwritten digit database," *ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist*, vol. 2, 2010. 7

[64] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Ng, "Reading digits in natural images with unsupervised feature learning," *NIPS*, 01 2011. 7

[65] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision (IJCV)*, vol. 115, no. 3, pp. 211–252, 2015. 7