FrequencyLowCut Pooling - Plug & Play against Catastrophic Overfitting

Julia Grabinski[1,2,3][0000−0002−8371−1734], Steffen Jung[4][0000−0001−8021−791X], Janis Keuper[2,3][0000−0002−1327−1243], and Margret Keuper[1,4][0000−0002−8437−7993]

[1] Visual Computing, Siegen University, Germany
[2] Competence Center High Performance Computing, Fraunhofer ITWM, Kaiserslautern, Germany
[3] Institute for Machine Learning and Analytics, Offenburg University, Germany
[4] Max Planck Institute for Informatics, Saarland Informatics Campus, Germany

Abstract. Over the last years, Convolutional Neural Networks (CNNs) have been the dominating neural architecture in a wide range of computer vision tasks. From an image and signal processing point of view, this success might be a bit surprising as the inherent spatial pyramid design of most CNNs is apparently violating basic signal processing laws, i.e. *Sampling Theorem* in their down-sampling operations. However, since poor sampling appeared not to affect model accuracy, this issue has been broadly neglected until model robustness started to receive more attention. Recent work [18] in the context of adversarial attacks and distribution shifts, showed after all, that there is a strong correlation between the vulnerability of CNNs and aliasing artifacts induced by poor down-sampling operations. This paper builds on these findings and introduces an aliasing free down-sampling operation which can easily be plugged into any CNN architecture: FrequencyLowCut pooling. Our experiments show, that in combination with simple and Fast Gradient Sign Method (FGSM) adversarial training, our hyper-parameter free operator substantially improves model robustness and avoids catastrophic overfitting. Our code is available at `https://github.com/GeJulia/flc_pooling`

Keywords: CNNs, Adversarial Robustness, Aliasing

1 Introduction

The robustness of convolutional neural networks has evolved to being one of the most crucial computer vision research topics in recent years. While state-of-the-art models provide high accuracy in many tasks, their susceptibility to adversarial attacks [9] and even common corruptions [20] is hampering their deployment in many practical applications. Therefore, a wide range of publications aim to provide models with increased robustness by adversarial training (AT) schemes [15,43,48], sophisticated data augmentation techniques [35] and enriching the training with additional data [4,16]. As a result, robuster models

can be learned with common CNN architectures, yet arguably at a high training cost - even without investigating the reasons for CNN's vulnerability. These reasons are of course multifold, starting with the high dimensionality of the feature space and sparse training data such that models easily tend to overfit [36,45]. Recently, the pooling operation in CNNs has been discussed in a similar context for example in [18] who measured the correlation between aliasing and a network's susceptibility to adversarial attacks. [50] have shown that commonly used pooling operations even prevent the smoothness of image representations under small input translations.

Our contributions are summarized as follows:

- We introduce FrequencyLowCut pooling, ensuring aliasing-free down-sampling within CNNs.
- Through extensive experiments with various datasets and architectures, we show empirically that FLC pooling prevents single step AT from catastrophic overfitting, while this is not the case for other recently published improved pooling operations (e.g. [50]).
- FLC pooling is substantially faster, around five times, and easier to integrate than previous AT or defence methods. It provides a hyperparameter-free plug and play module for increased model robustness.

1.1 Related Work

Adversarial Attacks. Adversarial attacks reveal CNNs vulnerabilities to intentional pixel perturbations which are crafted either having access to the full model (so-called white-box attacks) [15,33,42,33,3,27,37] or only having access to the model's prediction on given input images (so-called back-box attacks) [1,7]. The Fast Gradient Sign Method [15], FGSM, is an efficient single step white box attack. More effective methods use multiple optimization steps, e.g. as in the white-box Projected Gradient Descent (PGD) [27] or in black-box attacks such as Squares [1]. AutoAttack [9] is an ensemble of different attacks including an adaptive version of PGD and is widely used to benchmark adversarial robustness because of its strong performance [8]. In relation to image down-sampling, [47] and [30] demonstrate steganography-based attacks on the pre-processing pipeline of CNNs.

Adversarial Training. Some adversarial attacks are directly proposed with a dedicated defence [15,37]. Beyond these attack-specific defences, there are many methods for more general adversarial training (AT) schemes. These typically add an additional loss term which accounts for possible perturbations [12,48] or introduces additional training data [4,39]. Both are combined for example in [43], while [16] use data augmentation which is typically combined with weight averaging [35]. A widely used source for additional training data is *ddpm* [17,34,35], which contains one million extra samples for CIFAR-10 and is generated with the model proposed by [21]. [17] receive an additional boost in robustness by adding specifically generated images while [34] add wrongly labeled data to the training-set. RobustBench [8] gives an overview and evaluation of a variety of models w.r.t. their adversarial robustness and the additional data used.

A common drawback of all AT methods is the vast increase in computation needed to train networks: large amounts of additional adversarial samples and slower convergence due to the harder learning problem typically increase the training time by a factor between seven and fifteen [27,43,46,48].

Catastrophic Overfitting. AT with single step FGSM is a simple approach to achieve basic adversarial robustness [6,36]. Unfortunately, the robustness of this approach against stronger attacks like PGD is starting to drop again after a certain amount of training epochs. [45] called this phenomenon *catastrophic overfitting*. They concluded that one step adversarial attacks tend to overfit to the chosen adversarial perturbation magnitude (given by ϵ) but fail to be robust against multi-step attacks like PGD. [36] introduced early stopping as a countermeasure. After each training epoch, the model is evaluated on a small portion of the dataset with a multi-step attack, which again increases the computation time. As soon as the accuracy drops compared with a hand selected threshold the model training is stopped. [25] and [41] showed that the observed overfitting is related to the flatness of the loss landscape. They introduced a method to compute the *optimal* perturbation length ϵ' for each image and do single step FGSM training with this optimal perturbation length to prevent catastrophic overfitting. [2] showed that catastrophic overfitting not only occurs in deep neural networks but can also be present in single-layer convolutional neural networks. They propose a new kind of regularization, called GradAlign to improve FGSM perturbations and flatten the loss landscape to prevent catastrophic overfitting.

Anti-Aliasing. The problem of aliasing effects in the context of CNN-based neural networks has already been addressed from various angles in literature: [50] improve the shift-invariance of CNNs using anti-aliasing filters implemented as convolutions. [51] further improve shift invariance by using learned instead of predefined blurring filters. [29] rely on the low frequency components of wavelets during pooling operations to reduce aliasing and increase the robustness against common image corruptions. In [22] a depth adaptive blurring filter before pooling as well as an anti-aliasing activation function are used. Anti-aliasing is also relevant in the context of image generation. [24] propose to use blurring filters to remove aliases during image generation in generative adversarial networks (GANs) while [11] and [23] employ additional loss terms in the frequency space to address aliasing. In [18], we empirically showed via a proposed aliasing measure that adversarially robust models exhibit less aliasing in their down-sampling layers than non-robust models. Based on this motivation, we here propose an aliasing-free down-sampling operation that avoids catastrophic overfitting.

2 Preliminaries

2.1 Adversarial Training

In general, AT can be formalized as an optimization problem given by a *min-max* formulation:

$$\min_{\theta} \max_{\delta \in \Delta} L(x + \delta, y; \theta)\,, \tag{1}$$

where we seek to optimize network weights θ such that they minimize the loss L between inputs x and labels y under attacks δ. The maximization over δ can thereby be efficiently performed using the Fast Gradient Sign Method (FGSM), which takes one big step defined by ϵ into the direction of the gradient [15]:

$$x' = x + \epsilon \cdot \text{sign}(\nabla_x L(\theta, x, y)) \,. \tag{2}$$

Specific values of the perturbation size ϵ are usually set to be fractions of eight-bit encodings of the image color channels. A popular choice on the CIFAR-10 [26] dataset is $\epsilon = \frac{8}{255}$ which can be motivated by the human color perception [14]. The Projected Gradient Descent method, PGD, works similar to FGSM but instead of taking one big step in the direction of the gradient with step size ϵ, it iteratively optimizes the adversarial example with a smaller, defined step size α. Random restarts further increase its effectiveness. The final attack is clipped to the maximal step size of ϵ.

$$x'_{N+1} = \text{Clip}_{X,\epsilon}\{x'_N + \alpha \cdot \text{sign}(\nabla_x L(\theta, x, y))\} \tag{3}$$

PGD is one of the strongest attacks, due to its variability in step size and its random restarts. Yet, its applicability for AT is limited as it requires a relatively long optimization time for every example. Additionally, PGD is dependent on several hyperparameters, which makes it even less attractive for training in practice. In contrast, FGSM is fast and straight-forward to implement. Yet, models that use FGSM for AT tend to overfit on FGSM attacks and are not robust to other attacks such as PGD, i.e. they suffer from catastrophic overfitting [45].

2.2 Down-sampling in CNNs

Independent of their actual network topology, CNNs essentially perform a series of stacked convolutions and non-linearities. Using a vast amount of learnable convolution filters, CNNs are capable of extracting local texture information from all intermediate representations (input data and feature maps). To be able to abstract from this localized spatial information and to learn higher order relations of parts, objects and entire scenes, CNNs apply down-sampling operations to implement a spatial pyramid representation over the network layers.

This down-sampling is typically performed via a convolution with stride greater than one or by so-called pooling layers (see Fig. 1). The most common pooling layers are AveragePooling and MaxPooling. All of these operations are highly sensitive to small shifts or noise in the layer input [29,5,50].

Aliasing. Common CNNs sub-sample their intermediate feature maps to aggregate spatial information and increase the invariance of the network. However, no aliasing prevention is incorporated in current sub-sampling methods. Concretely, sub-sampling with too low sampling rates will cause pathological overlaps in the frequency spectra (Fig. 3). They arise as soon as the sampling rate is below the double bandwidth of the signal [40] and cause ambiguities: high frequency components can not be clearly distinguished from low frequency components. As a result, CNNs might misconceive local uncorrelated image perturbations

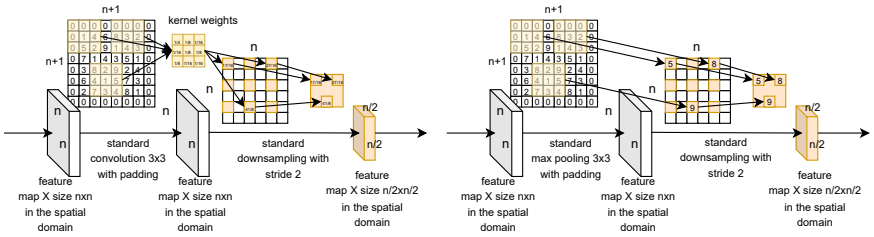

Fig. 1. Standard down-sampling operations used in CNNs. Left: down-sampling via convolution with stride two. First the feature map is padded and the actual convolution is executed. The stride defines the step-size of the kernel. Hence, for stride two, the kernel is moved two spatial units. In practice, this down-sampling is often implemented by a standard convolution with stride one and then discarding every second point in every spatial dimension. Right: down-sampling via MaxPooling. Here the max value for each spatial window location is chosen and the striding is implemented accordingly.

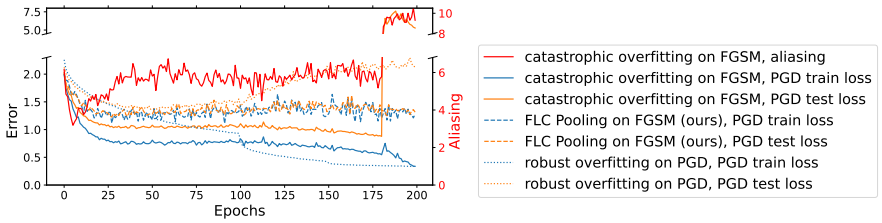

Fig. 2. Examples of AT facing catastrophic overfitting and its relationship to aliasing as well as robust overfitting and our FLC pooling. While FGSM training is prone to catastrophic overfitting, PGD training takes much longer and is also prone to robust overfitting. Our method, FLC pooling, is able to train with the fast FGSM training while preventing catastrophic overfitting.

as global manipulations. [18] showed that aliasing in CNNs strongly coincides with the robustness of the model. Based on this finding, one can hypothesize that models that overfit to high frequencies in the data tend to be less robust. This thought is also in line with the widely discussed texture bias [13]. To substantiate this hypothesis in the context of adversarial robustness, we investigate and empirically show in Figure 2 that catastrophic overfitting coincides with increased aliasing during FGSM AT. Based on this observation, we expect networks that sample without aliasing to be better behaved in AT FGSM settings. The FrequencyLowCut pooling, which we propose, trivially fulfills this property.

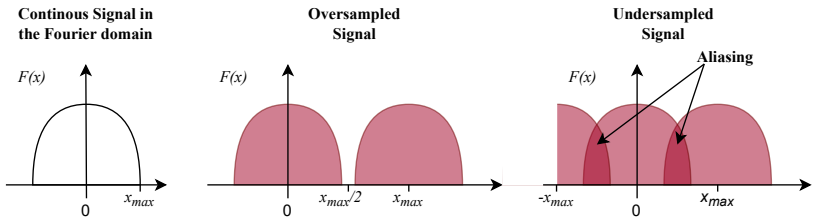

Fig. 3. Aliasing is apparent in the frequency domain. Left: The frequency spectrum of a 1D signal with maximal frequency $x_{\max}$. After down-sampling, replica of the signal appear at a distance proportional to the sampling rate. Center: The spectrum after sampling with a sufficiently large sampling rate. Right: The spectrum after under-sampling with aliases due to overlapping replica.

3 FrequencyLowCut Pooling

Several previous approaches such as [50,51] reduce high frequencies in features maps before pooling to avoid aliasing artifacts. They do so by classical blurring operations in the spatial domain. While those methods reduce aliasing, they can not entirely remove it due to sampling theoretic considerations in theory and limited filter sizes in practice (see Appendix A.3 or [14] for details). We aim to perfectly remove aliases in CNNs' down-sampling operations without adding additional hyperparameters. Therefore, we directly address the down-sampling operation in the frequency domain, where we can sample according to the Nyquist rate, i.e. remove all frequencies above $\frac{\text{samplingrate}}{2}$ and thus discard aliases. In practice, the proposed down-sampling operation first performs a Discrete Fourier Transform (DFT) of the feature maps f. Feature maps with height M and width N to be down-sampled are then represented as

$$F(k,l) = \frac{1}{MN} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f(m,n) e^{-2\pi j \left(\frac{k}{M} m + \frac{l}{N} n\right)} . \tag{4}$$

In the resulting frequency space representation F (eq. (4)), all frequencies k, l, with $|k|$ or $|l| > \frac{\text{samplingrate}}{2}$ have to be set to 0 before down-sampling. CNNs commonly down-sample with a factor of two, i.e. sampling rate $= \frac{1}{2}$. Down-sampling thus corresponds to finding $F_d(k,l) = F(k,l)\,, \forall\, \text{frequencies}\, k, l \,\text{with}\, |k|, |l| < \frac{1}{4}$. Practically, the DFT($f$) returns an array F of complex numbers with size $K \times L = M \times N$, where the frequency $k, l = 0$ is stored in the upper left corner and the highest frequency is in the center. We thus shift the low frequency components into the center of the array via FFT-shift to get F_s and crop the frequencies below the Nyquist frequency as $\mathrm{F}_{sd} = \mathrm{F}_s[K' : 3K', L' : 3L']$ for $K' = \frac{K}{4}$ and $L' = \frac{L}{4}$, for all samples in a batch and all channels in the feature map. After the inverse FFT-shift, we obtain array F_d with size $[\hat{K}, \hat{L}] = [\frac{K}{2}, \frac{L}{2}]$, containing exactly all frequencies below the Nyquist frequency F_d, which we can backtrans-

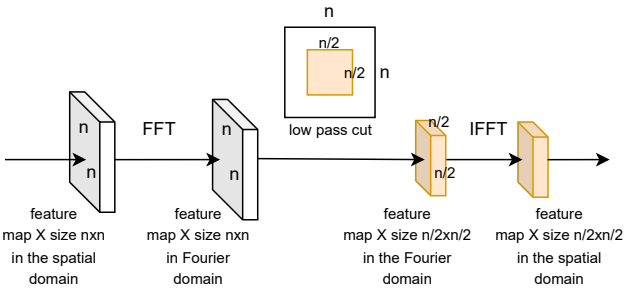

Fig. 4. FrequencyLowCut pooling, the proposed, guaranteed alias-free pooling operation. We first transform feature maps into frequency space via FFT, then crop the low frequency components. The result is transformed back into the spatial domain. This corresponds to a sinc-filtered and down-sampled feature map and is fed into the next convolutional layer.

form to the spatial domain via inverse DFT for the spatial indices $\hat{m} = 0 \dots \frac{M}{2}$ and $\hat{n} = 0 \dots \frac{N}{2}$.

$$f_d(\hat{m}, \hat{n}) = \frac{1}{\hat{K}\hat{L}} \sum_{k=0}^{\hat{K}-1} \sum_{l=0}^{\hat{L}-1} F_d(k, l) e^{2\pi j \left(\frac{\hat{m}}{\hat{K}} k + \frac{\hat{n}}{\hat{L}} l \right)}. \tag{5}$$

We thus receive the aliasing-free down-sampled feature map f_d with size $[\frac{M}{2}, \frac{N}{2}]$.

Fig. 4 shows this procedure in detail. In the spatial domain, this operation would amount to convolving the feature map with an infinitely large (non-bandlimited) $\text{sinc}(m) = \frac{\sin(m)}{m}$ filter, which can not be implemented in practice.

4 Experiments

4.1 Native Robustness of FLC pooling

We evaluate our proposed FLC pooling in a standard training scheme with Preact-ResNet-18 (PRN-18) architectures on CIFAR-10 (see Appendix A.1 for details). Table 1 shows that both the decrease in clean accuracy as well as the increase in robustness are marginal compared to the baseline models. We argue that these results are in line with our hypothesis that the removal of aliasing artifacts alone will not lead to enhanced robustness and we need to combine correct down-sampling with AT to compensate for the persisting problems induced by the very high dimensional decision spaces in CNNs.

4.2 FLC pooling for FGSM training

In the following series of experiments we apply simple FGSM AT with $\epsilon = \frac{8}{255}$ on different architectures and evaluate the resulting robustness with different

Table 1. Clean training of Preact-ResNet-18 (PRN-18) architectures on CIFAR-10. We compare clean and robust accuracy against FGSM [15] with L_{inf}, $\epsilon = \frac{8}{255}$, PGD [27] with L_{inf}, $\epsilon = \frac{1}{255}$ as well as L_2 with $\epsilon = 0.5$ (20 iterations) and common corruptions (CC) [20] (mean over all corruptions and severities).

Method	Clean	FGSM $\epsilon = \frac{8}{255}$	PGD L_{inf} $\epsilon = \frac{1}{255}$	PGD L_2 $\epsilon = 0.5$	CC
Baseline	**95.08**	34.08	7.15	6.68	74.38
FLC Pooling	94.66	**34.65**	**10.00**	**11.27**	**74.70**

Table 2. FGSM AT of PRN-18 and Wide-ResNet-28-10 (WRN-28-10) architectures on CIFAR-10. Comparison of clean and robust accuracy (high is better) against PGD [27] and AutoAttack [9] on the full dataset with L_{inf} with $\epsilon = 8/255$ and L_2 with $\epsilon = 0.5$. FGSM test accuracies indicate catastrophic overfitting on the AT data, hence this column is set to gray.

Method	Clean	FGSM $\epsilon = \frac{8}{255}$	PGD L_{inf} $\epsilon = \frac{8}{255}$	AA L_{inf} $\epsilon = \frac{8}{255}$	AA L_2 $\epsilon = 0.5$	AA L_{inf} $\epsilon = \frac{1}{255}$
Preact-ResNet-18						
Baseline: FGSM training	**90.81**	90.37	0.16	0.00	0.01	53.10
Baseline & early stopping	82.88	61.71	11.82	3.76	17.44	72.95
BlurPooling [50]	86.24	78.36	1.33	0.06	1.96	66.88
Adaptive BlurPooling [51]	90.35	77.39	0.23	0.00	0.07	39.00
Wavelet Pooling [28]	85.02	64.16	12.13	5.92	19.65	10.08
FLC Pooling (ours)	84.81	58.25	**38.41**	**36.69**	**55.58**	**80.63**
WRN-28-10						
Baseline: FGSM training	86.67	83.64	1.64	0.09	1.47	59.39
Baseline & early stopping	82.29	56.36	31.26	28.54	46.03	76.87
Blurpooling [50]	91.40	89.44	0.22	0.00	0.00	38.45
Adaptive BlurPooling [51]	91.10	89.76	0.00	0.00	0.00	7.42
Wavelet Pooling [28]	**92.19**	90.85	0.00	0.00	0.00	10.08
FLC Pooling (ours)	84.93	53.81	**39.48**	**38.37**	**52.89**	**80.27**

pooling methods. We compare the models in terms of their clean, FGSM, PGD and AutoAttack accuracy, where the FGSM attack is run with $\epsilon = 8/255$, PGD with 50 iterations and 10 random restarts and $\epsilon = 8/255$ and $\alpha = 2/255$. For AutoAttack, we evaluate the standard L_{inf} norm with $\epsilon = 8/255$ and a smaller ϵ of $1/255$, as AutoAttack is almost too strong to be imperceptible to humans [31]. Additionally, we evaluate AutoAttack with L_2 norm and $\epsilon = 0.5$.

CIFAR-10. Table 2 shows the evaluation of a PRN-18 as well as a Wide-ResNet-28-10 (WRN-28-10) on CIFAR-10 [26]. For both network architectures, we observe that our proposed FLC pooling is the only method that is able to prevent catastrophic overfitting. All other pooling methods heavily overfit on the FGSM training data, achieving high robustness towards FGSM attacks, but fail to generalize towards PGD or AutoAttack. Our hyper-parameter free approach also outperforms early stopping methods which are additionally suffering from the difficulty that one has to manually choose a suitable threshold in order to maintain the best model robustness.

Table 3. FGSM AT on CINIC-10 for PRN-18 architectures. We compare clean and robust accuracy (higher is better) against PGD [27] as well as AutoAttack [9] on the full dataset with L_{inf} with $\epsilon = 8/255$ and L_2 with $\epsilon = 0.5$. FGSM test accuracies indicate catastrophic overfitting on the AT data, hence this column is set to gray.

Method	Clean	FGSM $\epsilon = \frac{8}{255}$	PGD L_{inf} $\epsilon = \frac{8}{255}$	AA L_{inf} $\epsilon = \frac{8}{255}$	AA L_2 $\epsilon = 0.5$	AA L_{inf} $\epsilon = \frac{1}{255}$
Baseline	87.46	58.83	1.31	0.12	1.55	55.21
Baseline & early stopping	82.79	42.58	27.55	30.76	50.28	**79.88**
Blurpooling [50]	87.13	54.16	1.29	0.20	4.68	70.56
Adaptive BlurPooling [51]	**90.21**	52.27	0.05	0.00	0.01	40.96
aWavelet Pooling [28]	88.81	64.16	1.76	0.12	3.38	66.61
FLC Pooling (ours)	82.56	38.39	**36.28**	**49.61**	**60.51**	78.50

Table 4. FGSM AT on CIFAR-100 for PRN-18 architectures. We compare clean and robust accuracy (higher is better) against PGD [27] and AutoAttack [9] on the full dataset with L_{inf} with $\epsilon = 8/255$ and L_2 with $\epsilon = 0.5$. FGSM test accuracies indicate robustness to training data, so this column is set to gray. Here, none of the models overfit, while FLC pooling still yields best overall robustness.

Method	Clean	FGSM $\epsilon = \frac{8}{255}$	PGD L_{inf} $\epsilon = \frac{8}{255}$	AA L_{inf} $\epsilon = \frac{8}{255}$	AA L_2 $\epsilon = 0.5$	AA L_{inf} $\epsilon = \frac{1}{255}$
Baseline	51.92	23.25	15.41	11.13	25.67	44.53
Baseline & early stopping	52.09	23.34	15.51	10.88	25.78	44.61
Blurpooling [50]	52.68	23.40	16.81	12.43	26.79	45.68
Adaptive BlurPooling [51]	52.08	9.77	18.68	6.05	11.32	21.04
Wavelet Pooling [28]	**55.08**	25.70	18.36	13.76	**27.51**	47.52
FLC Pooling (ours)	54.66	26.82	**19.83**	**15.40**	26.30	**47.83**

CINIC-10. Table 3 shows similar results on CINIC-10 [10]. Our model exhibits no catastrophic overfitting, while previous pooling methods do. It should be noted that CINIC-10 is not officially reported by AutoAttack. This might explain why the accuracies under AutoAttack are higher on CINIC-10 than on CIFAR-10. We assume that AutoAttack is optimized for CIFAR-10 and CIFAR-100 and therefore less strong on CINIC-10.

CIFAR-100. Table 4 shows the results on CIFAR-100 [26], using the same experimental setup as for CIFAR-10 in Table 2. Due to the higher complexity of CIFAR-100, with ten times more classes than CIFAR-10, AT tends to suffer from catastrophic overfitting much later (in terms of epochs) in the training process. Therefore we trained the Baseline model for 300 epochs. While the gap towards the robustness of other methods is decreasing with the amount of catastrophic overfitting, our method still outperforms other pooling approaches in most cases - especially on strong attacks.

ImageNet. Table 5 evaluates our FLC Pooling on ImageNet. We compare against results reported on RobustBench [8], with emphasis on the model by [45] which also uses fast FGSM training. The clean accuracy of our model using FLC pooling is about 8% better than the one reached by [45], with a 1%

Table 5. Comparison of ResNet-50 models clean and robust accuracy against AutoAttack [9] on ImageNet. We compare against models reported on RobustBench [8].

Method	Clean	PGD L_{inf} $\epsilon = \frac{4}{255}$
Standard [8]	76.52	0.00
FGSM & FLC Pooling (ours)	63.52	27.29
Wong et al., 2020 [45]	55.62	26.24
Robustness lib, 2019 [12]	62.56	29.22
Salman et al., 2020 [38]	64.02	34.96

improvement in robust accuracy. All other models are trained with more time consuming methods like PGD (more details can be found in the Appendix A.2).

Analysis. The presented experiments on several datasets and architectures show that baseline FGSM training, as well as other pooling methods, strongly overfit on the adversarial data and do not generalize their robustness towards other attacks. We also show that our FLC pooling sufficiently prevents catastrophic overfitting and is able to generalize robustness over different networks, datasets, and attack sizes in terms of different ϵ-values.
Attack Structures. In Figure 5, we visualize AutoAttack adversarial attacks. Perturbations created for the baseline trained with FGSM differ substantially from those created for FLC pooling trained with FGSM. While perturbations for the baseline model exhibit high frequency structures, attacks to FLC pooling rather affect the global image structure.

4.3 Training Efficiency

Most AT approaches use adversarial image perturbations during training [15,27] [45]. Thereby the time and memory needed depend highly on the specific attack used to generate the perturbations. Multi-step attacks like PGD [27] require substantially more time than single step attacks like FGSM [45]. TRADES [48] incorporates different loss functions to account for a good trade-off between clean and robust accuracy. With our FLC pooling, we provide a simple and fast method for more robust models. Therefore we compare our method with state-of-the-art training schedules in terms of time needed per epoch when trained in their most basic form. Table 6 shows that FGSM training is fastest. However, FGSM with early stopping is not able to maintain high robustness against AutoAttack [9] due to catastrophic overfitting. PGD training can establish robustness against AutoAttack. It relies on the same training procedure as FGSM but uses expensive multi-step perturbations and thereby increases the computation time by over a factor of four (4.23). For Adversarial Weight Perturbations (AWP) the training time per epoch is over six times (6.57), for TRADES by eight times (8.04) higher. Our FLC pooling increases the training only by a factor of 1.26 while achieving a good clean and robust accuracy. When adding additional data like the *ddpm* dataset to the training as it is done in all leading RobustBench [8] models, the

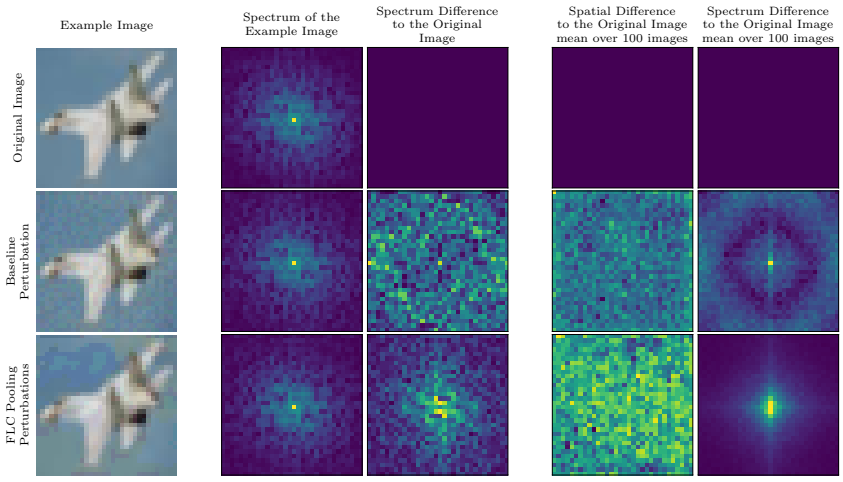

Fig. 5. Spatial and spectral differences of adversarial perturbations created by AutoAttack with $\epsilon = \frac{8}{255}$ on the baseline model as well as our FLC Pooling. On the left side for one specific example of an airplane and on the right side the average difference over 100 images.

training time is increased by a factor of twenty. The *ddpm* dataset incorporates one million extra samples, which is over sixteen times more than the original CIFAR-10 dataset. We report our training times for ImageNet in Appendix A.2 to show that FLC pooling is scalable in terms of practical runtime.

4.4 Black Box Attacks

PGD and AutoAttack are intrinsically related to FGSM. Therefore, to allow for a clean evaluation of the model robustness without bias towards the training scheme, we also evaluate black box attacks. Squares [1], which is also part of the AutoAttack pipeline, adds perturbations in the form of squares onto the image until the label flips. Besides Squares, we evaluate two transferred perturbations. The first perturbation set is constructed through the baseline network which is not robust at all. The second set is constructed from the baseline network which is trained with FGSM and early stopping. We evaluate against different PRN-18 and WRN-28-10 models on CIFAR-10 as well as PRN-18 models on CIFAR-100 provided by RobustBench [8]. Note that all networks marked with * are models which rely on additional data sources such as *ddmp* [21]. Other RobustBench models like [17] rely on training data that is not available anymore such that fair comparison is currently not possible. Arguably, we always expect models to further improve as training data is added.
Table 7 shows that for PRN-18 models our FLC pooling is consistently able to prevent black box attacks better while maintaining clean accuracy compared to other robust models from RobustBench. For WRN-28-10 models, we see a clear

Table 6. Runtime of AT in seconds per epoch over 200 epochs and a batch size of 512 trained with a PRN-18 for training on the original CIFAR-10 dataset without additional data. Experiments are executed on one Nvidia Tesla V100. Evaluation for clean and robust accuracy, higher is better, on AutoAttack [9] with our trained models. The models reported by the original authors may have different numbers due to different hyperparameter selection. The top row reports the baseline without AT.

Method	Seconds per epoch (avg)	Clean Acc	AA Acc
Baseline	14.6 ± 0.1	95.08	0.00
FGSM & early stopping [45]	**27.3 ± 0.1**	82.88	11.82
FGSM & FLC Pooling (Ours)	34.5 ± 0.1	**84.81**	38.41
PGD [27]	115.4 ± 0.2	83.11	40.35
Robustness lib [12]	117 ± 19.0	76.37	32.10
AWP [46]	179.4 ± 0.4	82.61	**49.43**
MART [44]	180.4 ± 0.8	55.49	8.63
TRADES [48]	219.4 ± 0.5	81.49	46.91

trend that models trained with additional data can achieve higher robustness. This is expected as wider networks can leverage additional data more effectively. One should note that all of these methods require different training schedules which are at least five times slower than ours and additional data which further increases the training time. For example, incorporating the *ddpm* dataset into the training increases the amount of training time by a factor of twenty. For CIFAR-100 (Table 8) our model is on par with [36].

4.5 Corruption Robustness

To demonstrate that our model generalizes the concept of robustness beyond adversarial examples, we also evaluate it on common corruptions incorporated with CIFAR-C [19]. We compare our model against our baseline as well as other RobustBench [8] models. Similar to the experiments on black box adversarial attacks we distinguish between models using only CIFAR-10 training data and models using extra-data like *ddpm* (marked by *). Table 7 shows that our FLC pooling, when trained only on CIFAR-10, can outperform other adversarially robust models as well as the baseline in terms of robustness against common corruptions for the PRN-18 architecture. As discussed above, WRN-28-10 models are designed to efficiently leverage additional data. As our model is exclusively trained on the clean CIFAR-10 dataset we can not establish the same robustness as other methods on wide networks. However, we can also see a substantial boost in robustness. Table 8 reports the results for CIFAR-100. There we can see that FLC pooling not only boosts clean accuracy but also robust accuracy on common corruptions.

4.6 Shift-Invariance

Initially, anti-aliasing in CNNs has also been discussed in the context of shift-invariance [50]. Therefore, after evaluating our model against adversarial and

Table 7. Robustness against black box attacks on PRN-18 and WRN-28-10 models with CIFAR-10. First against Squares [1] with $\epsilon = 1/255$ and then against perturbations which were created on the baseline network, meaning transferred perturbations (TP), and the baseline model including early stopping (TPE). As well as the accuracy under common corruptions (CC).

Model	Clean	Squares	TP	TPE	CC
Preact-ResNet-18					
Baseline	**90.81**	78.04	0.00	69.33	71.81
FGSM & early stopping	82.88	77.58	77.67	3.76	71.80
FGSM & FLC Pooling (ours)	84.81	**81.40**	**83.64**	**80.49**	**76.15**
Andriushchenko and Flammarion, 2020 [2]	79.84	76.78	78.65	75.06	72.05
Wong et al., 2020 [45]	83.34	80.25	82.03	78.81	74.60
Rebuffi et al., 2021 [35] *	83.53	81.24	82.36	80.28	75.79
WRN-28-10					
Baseline	86.67	76.17	0.09	67.3	77.33
FGSM & early stopping	82.29	78.01	80.8	28.54	72.55
FGSM & FLC Pooling (ours)	84.93	81.06	83.85	72.56	75.44
Carmon et al., 2019[4] *	**89.69**	**87.70**	**89.12**	**83.55**	81.30
Hendrycks et al., 2019 [20]	87.11	85.02	86.47	80.12	85.02
Wang et al., 2020 [44] *	87.50	85.30	86.74	80.65	**85.30**
Zhang et al., 2021 [49]	89.36	87.45	88.70	83.08	80.11

Table 8. Robustness against black box attacks for PRN-18 on CIFAR-100. First against Squares [1] with $\epsilon = 1/255$ and then against perturbations which were created on the baseline network, meaning transferred perturbations (TP), and the baseline model including early stopping (TPE). As well as the accuracy under common corruptions (CC).

Model	Clean	Squares	TP	TPE	CC
Baseline	51.92	45.74	11.13	23.91	41.22
FGSM & early stopping	52.09	45.75	23.90	10.88	41.15
FGSM & FLC Pooling (ours)	**54.66**	48.85	45.59	45.31	**44.18**
Rice et al., 2020 [36]	53.83	**48.92**	**45.97**	**46.11**	43.48

common corruptions, we also analyze its behavior under image shifts. We compare our model with the baseline as well as the shift-invariant models from [50] and [51].

FLC pooling can outperform all these specifically designed approaches in terms of consistency under shift, while BlurPooling [50] does not outperform the baseline. We assume that BlurPooling is optimized for larger image sizes like ImageNet, 224 by 224 pixels, compared to 32 by 32 pixels for CIFAR-10. The adaptive model from [51] is slightly better than the baseline but can not reach the consistency of our model.

5 Discussion & Conclusions

The problem of aliasing in CNNs or GANs has recently been widely discussed [11,23,24]. We contribute to this field by developing a fully aliasing-free down-

Table 9. Consistency of PRN-18 model prediction under image shifts on CIFAR-10. Each model is trained without AT with the same training schedule (see Appendix A.1 for details).

Model	Clean	Consistency under shift
Baseline	94.78	86.48
BlurPooling [50]	**95.04**	86.19
adaptive BlurPooling [51]	94.97	91.47
FLC Pooling (ours)	94.66	**94.46**

sampling layer that can be plugged into any down-sampling operation. Previous attempts in this direction are based on blurring before down-sampling. This can help to reduce aliasing but can not eliminate it. With FLC pooling we developed a hyperparameter-free and easy plug-and-play down-sampling which supports CNNs native robustness. Thereby, we can overcome the issue of catastrophic overfitting in single-step AT and provide a path to reliable and fast adversarial robustness. We hope that FLC pooling will be used to evolve to fundamentally improved CNNs which do not need to account for aliasing effects anymore.

References

1. Andriushchenko, M., Croce, F., Flammarion, N., Hein, M.: Square attack: a query-efficient black-box adversarial attack via random search. In: European Conference on Computer Vision. pp. 484–501. Springer (2020)
2. Andriushchenko, M., Flammarion, N.: Understanding and improving fast adversarial training. Advances in Neural Information Processing Systems **33**, 16048–16059 (2020)
3. Carlini, N., Wagner, D.: Towards evaluating the robustness of neural networks. In: 2017 ieee symposium on security and privacy (sp). pp. 39–57. IEEE (2017)
4. Carmon, Y., Raghunathan, A., Schmidt, L., Duchi, J.C., Liang, P.S.: Unlabeled data improves adversarial robustness. Advances in Neural Information Processing Systems **32** (2019)
5. Chaman, A., Dokmanic, I.: Truly shift-invariant convolutional neural networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3773–3783 (2021)
6. Chen, T., Zhang, Z., Liu, S., Chang, S., Wang, Z.: Robust overfitting may be mitigated by properly learned smoothening. In: International Conference on Learning Representations (2021), `https://openreview.net/forum?id=qZzy5urZw9`
7. Cheng, M., Le, T., Chen, P.Y., Yi, J., Zhang, H., Hsieh, C.J.: Query-efficient hard-label black-box attack: An optimization-based approach. arXiv preprint arXiv:1807.04457 (2018)
8. Croce, F., Andriushchenko, M., Sehwag, V., Flammarion, N., Chiang, M., Mittal, P., Hein, M.: Robustbench: a standardized adversarial robustness benchmark. arXiv preprint arXiv:2010.09670 (2020)
9. Croce, F., Hein, M.: Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In: ICML (2020)

10. Darlow, L.N., Crowley, E.J., Antoniou, A., Storkey, A.J.: Cinic-10 is not imagenet or cifar-10. arXiv preprint arXiv:1810.03505 (2018)
11. Durall, R., Keuper, M., Keuper, J.: Watch your up-convolution: Cnn based generative deep neural networks are failing to reproduce spectral distributions (2020)
12. Engstrom, L., Ilyas, A., Salman, H., Santurkar, S., Tsipras, D.: Robustness (python library) (2019), https://github.com/MadryLab/robustness
13. Geirhos, R., Rubisch, P., Michaelis, C., Bethge, M., Wichmann, F.A., Brendel, W.: Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. arXiv preprint arXiv:1811.12231 (2018)
14. Gonzalez, R.C., Woods, R.E.: Digital Image Processing (3rd Edition). Prentice-Hall, Inc., USA (2006)
15. Goodfellow, I.J., Shlens, J., Szegedy, C.: Explaining and harnessing adversarial examples (2015)
16. Gowal, S., Qin, C., Uesato, J., Mann, T., Kohli, P.: Uncovering the limits of adversarial training against norm-bounded adversarial examples (2021)
17. Gowal, S., Rebuffi, S.A., Wiles, O., Stimberg, F., Calian, D.A., Mann, T.A.: Improving robustness using generated data. Advances in Neural Information Processing Systems **34** (2021)
18. Grabinski, J., Keuper, J., Keuper, M.: Aliasing coincides with CNNs vulnerability towards adversarial attacks. In: The AAAI-22 Workshop on Adversarial Machine Learning and Beyond (2022), https://openreview.net/forum?id=vKc1mLxBebP
19. Hendrycks, D., Dietterich, T.: Benchmarking neural network robustness to common corruptions and perturbations. Proceedings of the International Conference on Learning Representations (2019)
20. Hendrycks, D., Lee, K., Mazeika, M.: Using pre-training can improve model robustness and uncertainty. In: International Conference on Machine Learning. pp. 2712–2721. PMLR (2019)
21. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. Advances in Neural Information Processing Systems **33**, 6840–6851 (2020)
22. Hossain, M.T., Teng, S.W., Sohel, F., Lu, G.: Anti-aliasing deep image classifiers using novel depth adaptive blurring and activation function (2021)
23. Jung, S., Keuper, M.: Spectral distribution aware image generation. In: AAAI (2021)
24. Karras, T., Aittala, M., Laine, S., Härkönen, E., Hellsten, J., Lehtinen, J., Aila, T.: Alias-free generative adversarial networks. Advances in Neural Information Processing Systems **34** (2021)
25. Kim, H., Lee, W., Lee, J.: Understanding catastrophic overfitting in single-step adversarial training (2020)
26. Krizhevsky, A.: Learning multiple layers of features from tiny images. University of Toronto (05 2012)
27. Kurakin, A., Goodfellow, I., Bengio, S.: Adversarial machine learning at scale (2017)
28. Li, Q., Shen, L., Guo, S., Lai, Z.: Wavelet integrated cnns for noise-robust image classification. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 7245–7254 (2020)
29. Li, Q., Shen, L., Guo, S., Lai, Z.: Wavecnet: Wavelet integrated cnns to suppress aliasing effect for noise-robust image classification. IEEE Transactions on Image Processing **30**, 7074–7089 (2021). https://doi.org/10.1109/tip.2021.3101395, http://dx.doi.org/10.1109/TIP.2021.3101395
30. Lohn, A.J.: Downscaling attack and defense: Turning what you see back into what you get (2020)

31. Lorenz, P., Strassel, D., Keuper, M., Keuper, J.: Is robustbench/autoattack a suitable benchmark for adversarial robustness? In: The AAAI-22 Workshop on Adversarial Machine Learning and Beyond (2022), https://openreview.net/forum?id=aLB3FaqoMBs
32. Madry, A., Makelov, A., Schmidt, L., Tsipras, D., Vladu, A.: Towards deep learning models resistant to adversarial attacks. arXiv preprint arXiv:1706.06083 (2017)
33. Moosavi-Dezfooli, S.M., Fawzi, A., Frossard, P.: Deepfool: a simple and accurate method to fool deep neural networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2574–2582 (2016)
34. Rade, R., Moosavi-Dezfooli, S.M.: Helper-based adversarial training: Reducing excessive margin to achieve a better accuracy vs. robustness trade-off. In: ICML 2021 Workshop on Adversarial Machine Learning (2021), https://openreview.net/forum?id=BuD2LmNaU3a
35. Rebuffi, S.A., Gowal, S., Calian, D.A., Stimberg, F., Wiles, O., Mann, T.: Fixing data augmentation to improve adversarial robustness (2021)
36. Rice, L., Wong, E., Kolter, Z.: Overfitting in adversarially robust deep learning. In: International Conference on Machine Learning. pp. 8093–8104. PMLR (2020)
37. Rony, J., Hafemann, L.G., Oliveira, L.S., Ayed, I.B., Sabourin, R., Granger, E.: Decoupling direction and norm for efficient gradient-based l2 adversarial attacks and defenses. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4322–4330 (2019)
38. Salman, H., Ilyas, A., Engstrom, L., Kapoor, A., Madry, A.: Do adversarially robust imagenet models transfer better? Advances in Neural Information Processing Systems **33**, 3533–3545 (2020)
39. Sehwag, V., Mahloujifar, S., Handina, T., Dai, S., Xiang, C., Chiang, M., Mittal, P.: Improving adversarial robustness using proxy distributions (2021)
40. Shannon, C.: Communication in the presence of noise. Proceedings of the IRE **37**(1), 10–21 (1949). https://doi.org/10.1109/JRPROC.1949.232969
41. Stutz, D., Hein, M., Schiele, B.: Relating adversarially robust generalization to flat minima (2021)
42. Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., Fergus, R.: Intriguing properties of neural networks. In: International Conference on Learning Representations (2014), http://arxiv.org/abs/1312.6199
43. Wang, Y., Zou, D., Yi, J., Bailey, J., Ma, X., Gu, Q.: Improving adversarial robustness requires revisiting misclassified examples. In: International Conference on Learning Representations (2020), https://openreview.net/forum?id=rklOg6EFwS
44. Wang, Y., Zou, D., Yi, J., Bailey, J., Ma, X., Gu, Q.: Improving adversarial robustness requires revisiting misclassified examples. In: International Conference on Learning Representations (2020), https://openreview.net/forum?id=rklOg6EFwS
45. Wong, E., Rice, L., Kolter, J.Z.: Fast is better than free: Revisiting adversarial training. In: International Conference on Learning Representations (2020), https://openreview.net/forum?id=BJx040EFvH
46. Wu, D., Xia, S.T., Wang, Y.: Adversarial weight perturbation helps robust generalization. Advances in Neural Information Processing Systems **33**, 2958–2969 (2020)
47. Xiao, Q., Li, K., Zhang, D., Jin, Y.: Wolf in sheep's clothing - the downscaling attack against deep learning applications (2017)

48. Zhang, H., Yu, Y., Jiao, J., Xing, E.P., Ghaoui, L.E., Jordan, M.I.: Theoretically principled trade-off between robustness and accuracy. In: International Conference on Machine Learning (2019)
49. Zhang, J., Zhu, J., Niu, G., Han, B., Sugiyama, M., Kankanhalli, M.: Geometry-aware instance-reweighted adversarial training. In: International Conference on Learning Representations (2021), https://openreview.net/forum?id=iAX0l6Cz8ub
50. Zhang, R.: Making convolutional networks shift-invariant again. In: ICML (2019)
51. Zou, X., Xiao, F., Yu, Z., Lee, Y.J.: Delving deeper into anti-aliasing in convnets. In: BMVC (2020)