Robustness to Perturbations in the Frequency Domain: Neural Network Verification and Certified Training

Harleen Hanspal
Imperial College London
h.hanspal21@imperial.ac.uk

Alessandro De Palma
Inria, École Normale Supérieure, PSL University, CNRS
alessandro.de-palma@inria.fr

Alessio Lomuscio
Imperial College London, Safe Intelligence UK
alessio@safeintelligence.ai

Abstract

Deploying neural networks in safety critical applications such as autonomous driving requires assurance on their robustness. Deterministic robustness assessment can be made using formal verification. Existing frameworks for network verification and certified training verify and robustify networks against specifications capturing specific transformations, or perturbations in the pixel or latent space. However, recent works highlight the vulnerability of networks to perturbations and attacks in the frequency domain, which cannot be precisely captured by the existing specifications. Therefore, we present a framework to encode, verify and robustly train for frequency-characterised specifications. Our approach defines input specifications in the Fourier domain and propagates them using an inverse Fourier-transform encoding network prepended to the network to be verified. We demonstrate the ability of our framework to encode perturbations across the spectrum, from the low-frequency intensity changes up to the high-frequency white noise, kernel-based and domain changes. We then use SoA verifiers to verify differently-trained networks for non-trivial robustness guarantees against some of these practically relevant specifications. Finally, we integrate our framework within existing certified training schemes to enhance network's verified robustness against the proposed specifications by up to 50%.

1. Introduction

The deployment of Neural Networks (NNs) for safety-critical applications, such as autonomous driving and aviation, requires assurance on both their accuracy and robustness. Network robustness can be formalised as an input-output constraint set or *specification*, and be deterministically guaranteed with *formal verification*. Typical input specifications for vision models are ℓ_p-norm balls in pixel space, such that verifying them provides a lower bound on network's worst-case performance against all possible norm-bounded pixel perturbations and attacks, irrespective of the attack algorithm.

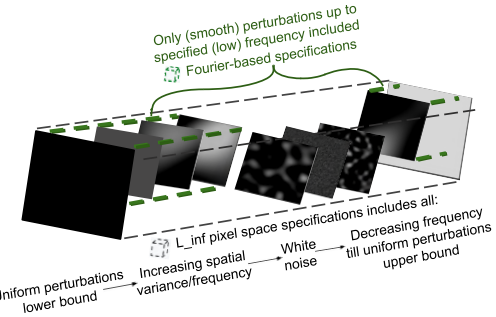


Figure 1. Depiction of frequency-based specifications offering greater precision in capturing intended perturbations than a pixel space ℓ_∞-specification for the same perturbation bounds.

The past few years have seen the development of frequency-based attacks [3, 18, 30, 35] that exploit NNs' uneven sensitivity to perturbations of different frequency characteristics [35]. The frequency domain and its transforms allow constructing perturbations with interdependence among pixels, such as with constraints on the intensity change among adjacent pixels. Thereby, one can devise low frequency attacks which appear like smooth illumination changes that are likely to naturally occur in the real world. These attacks are hard to detect-and-obfuscate [26], diverse [16, 18, 45], and efficient to devise given their low-dimensionality [12, 35] (discussed in §3.3). Since the typical pixel-space ℓ_∞-norm balls do not impose spatial constraints among pixels, they are unable to precisely capture these perturbations (see Fig1).

Considering the ubiquity and diversity of frequency-based attacks, and the fact that deterministic robustness against them has not been explored, in this work,

- we define input specifications in terms of Fourier coeffi-

cients to capture perturbation sets, and the attacks therein, as characterised by their constituent frequencies; the perturbations covered include diverse spatial intensity variations, kernel-based changes and domain/style changes,

- we present a framework that encodes these diverse perturbation classes and enables their verification and certified training by SoA ℓ_∞-norm-based tools,
- we verify differently trained networks for the proposed specifications to ascertain the absence of any network vulnerability or attack conforming to these specifications,
- we perform certified training for the proposed specifications as a deterministic certified defense against frequency-based attacks.

Broadly, we find that with the proposed approach, even non-robustly trained networks, which could not be verified for pixel-space specifications, could be verified for up to non-trivial accuracies for the low-frequency specifications. We also find that certified training with Forward+Backward bound propagation induces up to 21% higher accuracy and 50% higher verified robustness for our low-frequency specifications. Thereby, we believe that the frequency domain-based approach of certified training and verification is an apt candidate to support the pixel domain-based approaches for the reliability enhancement and assessment of NNs prior to their safety-critical real-world deployments. Our code is available at https://github.com/hh10/Fourier-Verification-and-Certified-Training.

Related Work NNs have been reported to being unevenly sensitive over the Fourier basis [30] and relying critically on low frequencies in images [41]. As such, there exist diverse Fourier-based attacks, varying in the frequency-bands attacked [18,45] and the algorithms (gradient-based [45], evolutionary algorithm-based [34]) used. The existing works on frequency domain-based augmentations [7, 31, 40], attacks and training [16,35] improve network's generalisation and empirical performance against frequency-based attacks. However, they cannot consider all examples conforming to the chosen perturbation class, and thus do not produce formal guarantees on network robustness. In contrast, this work encloses the perturbation set in its entirety to provide guarantees on network's response to the entire set and any attack therein. Some works develop frequency-based attacks [23,34] and defenses [3] in Discrete Cosine Transform (DCT) domain. The DCT basis are real-valued cosines with the phase subsumed in their magnitudes. In contrast, the DFT basis are complex exponentials maintaining explicit frequency amplitudes and phases. These amplitudes and phases are known to separately capture the visual style and structure of the image. Therefore, we primarily use DFT for specification design to support a wider set of perturbations; although, our approach (except the amplitude-phase magnitude relations) using DCT is identical and discussed in App.D.2. While some recent works proposed probabilistic certificates for robustness against frequency-based attacks [1,2], this work focuses on similar deterministic guarantees using formal verification.

Most existing works on verification and certified training of NNs verify and enhance their robustness to the high-dimensional pixel-domain specifications, which are configurable in terms of the perturbation magnitude. In contrast, our focus is the potentially low-dimensional frequency-domain specifications, which offer additional configurability of the frequency characteristics of the enclosed perturbations (Fig1). There exist a few works catering to low-dimensional perturbations such as (a) geometric [4], spatial [25], color-space transformations [21], (b) biasfields [14], and (c) learnt manifold-based [13, 20] perturbations. While their frameworks capture diverse perturbations, they are tied to specific transformations and basis, so not easily extendable in case of (a-b), not interpretable in terms of a physical property in case of (b), and reliant on a heuristic training process in case of (c). In contrast, the Fourier-based approach uses transforms which are neither learnt nor perturbation-specific, and the decoded perturbations can be interpreted in terms of their frequencies. In terms of the nature and implementation of the perturbations, the work most relevant to ours is (b) biasfields. The biasfields framework involves constructing a basis set of spatial polynomials of finite size or rank (see Fig2) and prepending these polynomials as network layers to the NN to be verified. The richness of their encoded perturbations depends on and is limited by the rank of their basis set.

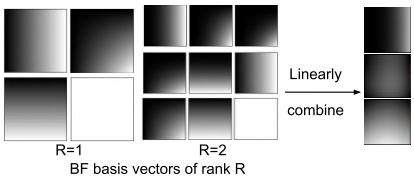


Figure 2. Biasfield (BF) perturbations δ are combinations of BF basis vectors $\{B_k\}_{k=0}^{R}$ of given rank R, i.e., $\delta := \sum_{k=0}^{R} a_k B_k$.

2. Background

NN verification and specifications Given a network $N_\theta : X \to Y$ with parameters θ and an input-output specification $(X_{\text{des}}, Y_{\text{des}})$, the network verification problem is the decision problem determining whether $N_\theta(x) \in Y_{\text{des}}, \forall x \in X_{\text{des}}$. A positive verification outcome guarantees that there is no potential attack $x \in X_{\text{des}}$ such that $N_\theta(x) \notin Y_{\text{des}}$. The input specification X_{des} for an image x is typically defined as:

1. *Pixel space sets*, such that $X_{\text{des}}(x)$ is an ℓ_p [17] or Wasserstein [36] norm ball around x, i.e., $\mathcal{B}_\epsilon(x) = \{x' \mid d(x', x) < \epsilon\}$. $X_{\text{des}}(x)$ captures norm-bounded attacks, and photometric and geometric changes [4] although with non-conforming samples. Henceforth, we denote the bounding magnitude for norm-balls in ℓ_p norm by ϵ_p.
2. *Transformation parameters*, such that $X_{\text{des}}(x, \alpha)$ is de-

fined by prepending transform layers $N_T(\alpha)$ to N_θ, with $\alpha \in \mathcal{B}_\epsilon(0)$ determining the transformation extent. The $X_{\text{des}} := N_T(\alpha)(x)$, which based on N_T's definition, can capture geometric, color [21], and smooth intensity [14] changes more precisely.

For classification tasks, Y_{des} encodes prediction accuracy, and *verified accuracy* denotes the ratio of inputs $x \in X$ for which network prediction is correct for the entire $X_{\text{des}}(x)$.

NN verifiers Several verification methods, such as based on SMT [17], MILP [10], Bound Propagation (BP) [29, 32, 33, 44], SDP [24], Lipschitzness [44], can be used to verify NNs. Most of these methods determine NN's reachable output set for an input *set* $X_{\text{des}} : \bar{Y}_{\text{des},\theta} \supseteq \{N_\theta(x) \mid x \in X_{\text{des}}\}$. If $\bar{Y}_{\text{des},\theta} \subseteq Y_{\text{des}}$, then the property being verified is said to be satisfied. Most BP-based verifiers over-approximate network outputs, and are therefore *incomplete*, i.e., they may not always establish the satisfiability of a property when it holds. For complete verification, a search routine within Branch-and-Bound (BaB) is used on top of BP [6, 8, 33, 39]. Being expensive, BaB is used only for verification and not training. Verifiers used in this work use the following BP methods listed in order of typically increasing precision and computational overhead:

- Interval (IBP) [11]: each layer's bounds are axes-aligned intervals, depend on its own input's bounds and mapping,
- Forward [42]: each layer's bounds are linear in terms of NN's input bounds and depend on all preceding layers,
- Backward [43]: given pre-computed bounds of all layers, each layer's bounds are propagated in reverse in terms of NN's output bounds, and depend on all succeeding layers.

A low input/output dimensionality is critical to make Forward/Backward BP feasible for use in training [14].

Certified training Given a standard loss function L for a learning task, a natural way to train a network for robustness as per an input specification X_{des} is via the minimax robust loss: $\min_\theta \mathbb{E}_{(x,y)\in(X,Y)}\left[\max_{x'\in X_{\text{des}}} L(N_\theta(x'), y)\right]$. Owing to the infeasibility of the inner maximization, this worst-case loss is approximated. A lower bound-approximation lends to adversarial training [19], yielding networks with empirical but not verifiable robustness. An upper-bound approximation, as using the over-approximations made by the incomplete verifiers, lends to certified training [9, 11, 22, 28, 37, 43] and yields networks amenable to formal verification.

Frequency domain and transforms Frequency domain-based analysis interprets data in terms of its constituent frequencies. A popular way to obtain the frequency content of digital signals is Discrete Fourier Transform (DFT).

Relevant notation. We denote complex numbers and frequency-domain entities by capped symbols, such as $\hat{x} := |\hat{x}|e^{\angle\hat{x}}$ *(polar form)* $= \Re(\hat{x}) + \iota\Im(\hat{x})$ *(Cartesian form)*, where $|\hat{x}|, \angle\hat{x}, \Re(\hat{x}), \Im(\hat{x})$ denote their amplitude, phase, real and imaginary components. $\|.\|_p, \cdot$ and $\odot$ denote the ℓ_p norm, dot product and element-wise product respectively.

The DFT decomposes data into harmonics of the fundamental frequency ω_f, which is a function of the data length. We define DFT $\mathcal{F} : \mathcal{X} \in \mathbb{R}^D \to \hat{\mathcal{X}} \in \mathbb{C}^D$ and its inverse (IDFT) $\bar{\mathcal{F}}^1$ for $2d$ data as:

$$\hat{x}[k,l] = |\hat{x}[k,l]|e^{\angle\hat{x}[k,l]} = \frac{1}{WH}\sum_{n=0}^{H-1}\sum_{m=0}^{W-1} x[m,n]e^{-\iota 2\pi\left(\frac{kn}{H}+\frac{lm}{W}\right)} \quad (1)$$

$$x[m,n] = \sum_{k=0}^{H-1}\sum_{l=0}^{W-1} \hat{x}[k,l]e^{\iota 2\pi\left(\frac{kn}{H}+\frac{lm}{W}\right)},$$

where $x \in X, D = H \times W$. The above can be computed as a matrix multiplication using the (I)DFT matrices ${\Omega^{(1)}}^{-1}_{HW}$. The FT properties [5] used in this work are: linearity $\mathcal{F}(ax + by) = a\mathcal{F}(x) + b\mathcal{F}(y),\ a, b \in \mathbb{R}$; circular convolution $\mathcal{F}(x \circledast y) = \mathcal{F}(x) \times \mathcal{F}(y)$; conjugate-symmetricity of DFT of a real signals; Complex multiplication property $\hat{x}_1\hat{x}_2 = (\Re(\hat{x}_1)\Re(\hat{x}_2) - \Im(\hat{x}_1)\Im(\hat{x}_2)) + \iota(\Im(\hat{x}_1)\Re(\hat{x}_2) + \Re(\hat{x}_1)\Im(\hat{x}_2))$, and the Parseval's theorem. Typically, smooth data can be encoded using a few low frequencies, while data with high spatial variance requires more frequencies. However, since the DFT basis are infinite and periodic, DFT analysis can require a finite-length data to be periodised, i.e., infinitely repeated with a period D equal to its length. On periodisation, even smooth data, if not periodic with period D, could become discontinuous and have high frequencies in its DFT leading to *aliasing*.

3. Fourier Domain Specification Encoding

This section presents the proposed Fourier-based input specifications, their enclosing of existing frequency-domain attacks, and the encoding network that enables their incorporation in the existing SoA verification and certified training pipelines.

Formally, given a network N_θ and a specification $(X_{\text{des}}, Y_{\text{des}})$, where X_{des} is a discrete domain, we define the Fourier Domain Verification Problem (FDVP) as the decision problem determining whether $N_\theta(x) \in Y_{\text{des}}, \quad \forall x \in X_{\text{des}}$, where X_{des} supports the following Fourier domain-based input specification classes:

$$X_{\text{des,add}}(\hat{\delta}) := \{x + \bar{\mathcal{F}}^1(\hat{\delta}) \mid \hat{\delta} \in \mathcal{B}_{p,\hat{\epsilon},\hat{M}}(\mathbf{0})\} \quad (2)$$

$$X_{\text{des,mul}}(\hat{\delta}) := \{x \odot \left(1 + \bar{\mathcal{F}}^1(\hat{\delta})\right) \mid \hat{\delta} \in \mathcal{B}_{p,\hat{\epsilon},\hat{M}}(\mathbf{0})\} \quad (3)$$

$$X_{\text{des,conv}}(\hat{\delta}) := \{\bar{\mathcal{F}}^1(\mathcal{F}(x) \odot \hat{\delta}) \mid \hat{\delta} \in \mathcal{B}_{p,\hat{\epsilon},\hat{M}}(\mathcal{F}(K))\} \quad (4)$$

$$X_{\text{des,cond}}(\alpha) := \left\{ \begin{array}{l} \bar{\mathcal{F}}^1\left((\alpha|\mathcal{F}(x)| + (1-\alpha)|\mathcal{F}(x')|)\, e^{\angle\mathcal{F}(x)}\right) \\ \mid \alpha \in [0,1], (x,x') \in (X,X') \end{array} \right\} \quad (5)$$

In the above, the $\mathcal{B}_{p,\hat{\epsilon},\hat{M}}$ is a norm-bounded perturbation ball defined as $\mathcal{B}_{p,\hat{\epsilon},\hat{M}}(\hat{\delta}) := \{\hat{\delta}' \mid \|\hat{M} \odot (\hat{\delta}' - \hat{\delta})\|_p \le \hat{\epsilon}\}$, the $\hat{M} \in \{0,1\}^{|\mathcal{F}(X)|}$ is a mask that governs the Fourier coefficients being perturbed, and the K is the pixel domain convolution kernel. Equations (2), (3) and (4) define single input-based specifications that encode the additive, multiplicative and convolution-based perturbations respectively. Equation (5) defines a two input-based conditional specification intended to capture the visual style change between

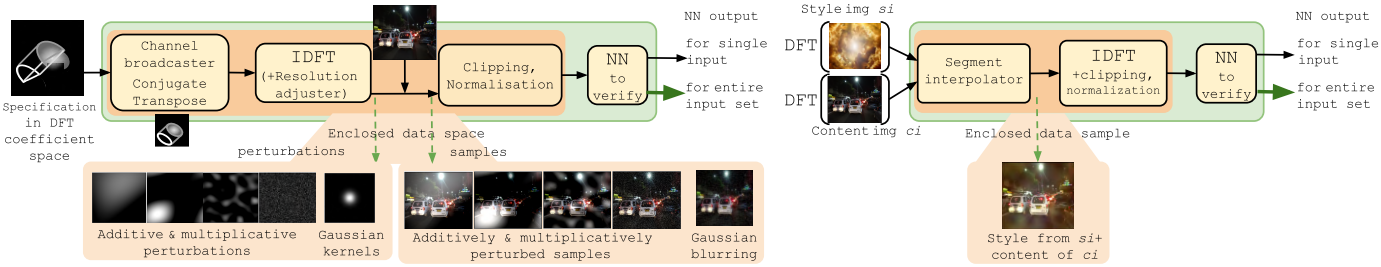


Figure 3. The proposed pipelines for verification of DFT domain-based specifications. The left and right pipelines support the single-input zero-centered additive, multiplicative and the convolution-based perturbations, and the two input-based conditional specifications resp. In both, the orange and green blocks show our encoding network and the complete verification path resp.

two input spaces X and X' while retaining the structural content from X in the enclosed perturbations. Details and representative examples for each of these specifications are provided in §4.1, particularly in Fig4.

The IDFT operation $\bar{\mathcal{F}}^1$ transforms the ℓ_p-norm ball in DFT domain, $\mathcal{B}_{p,\hat{\epsilon},\hat{M}}(\hat{\delta})$, back to the pixel domain so that the vision-based networks, that typically operate on the pixel domain, can be tested for these specification without any modification. Next, we define the encoding network that implements our specifications X_{des} (2)-(5) using standard network layers.

3.1. Encoding Network

Our encoding framework is shown in Fig 3. Encoding single input specifications involves defining a zero-centered perturbation set $\mathcal{B}_{p,\hat{\epsilon},\hat{M}}(0)$ in the DFT coefficient space, soundly propagating it through the IDFT network to get the corresponding perturbation set in the pixel space, centering this set around an input image x to get the locally-perturbed input set, and finally propagating it through the NN to be verified. Therefore, the specification encoding network E for the single input perturbations (2)-(4) can be written as:

$$E(\hat{\zeta}, x) = \text{CN} \circ \text{PM}(x) \circ \text{IDFT}(\hat{\zeta}).$$

Encoding two-input conditional specifications involves encoding a scalar-parameterised path between the Fourier coefficients $(\hat{\zeta}_1, \hat{\zeta}_2')$ of two images as per (5), and propagating this path through IDFT and the NN to be verified. This encoding network can be written as:

$$E(\hat{\zeta}_1, \hat{\zeta}_2') = \text{CN} \circ \text{IDFT} \circ \text{SI}(\hat{\zeta}_1, \hat{\zeta}_2').$$

The inputs to both E are Complex-valued Fourier coefficients $\hat{\delta} \in \mathbb{C}^D$ as a stacked vector of their Cartesian components, i.e, $\hat{\zeta} := \begin{bmatrix} \Re(\hat{\delta}) \\ \Im(\hat{\delta}) \end{bmatrix} \in \mathbb{R}^{2D}$. The main modules of E are briefly described below; their details and the optional modules are deferred to App.A.

IDFT. This module transforms the Fourier domain inputs $\hat{\zeta}$ to pixel domain samples $\delta \in \mathbb{R}^D$, using the Complex multiplication property, as follows:

$$\delta = \Re(\Omega_N^1 \hat{\zeta} \Omega_N^{1T}) = \begin{bmatrix} \Re(\Omega_N^1) & -\Im(\Omega_N^1) \\ \Im(\Omega_N^1) & \Re(\Omega_N^1) \end{bmatrix} \begin{bmatrix} \Re(\hat{\delta}) \\ \Im(\hat{\delta}) \end{bmatrix} \cdot \begin{bmatrix} \Re(\Omega_N^1) \\ -\Im(\Omega_N^1) \end{bmatrix}.$$

Since the SoA verifiers do not support Complex-valued layers, we implement IDFT as above, with each matrix multiplication done using real-valued linear layers. The DFT coefficients are normalized as in equation (1) and the typical FT shift operations are avoided to have smaller input bounds and fewer operations in the verification path resp.

PM. The PM module centers the pixel domain perturbation set around an image batch as per (2)-(4).

SI. Given two Fourier coefficients $(\hat{\zeta}_1, \hat{\zeta}_2)$, the Segment Interpolator module takes a scalar input $\alpha \in [0, 1]$ and outputs their interpolations, i.e., $\hat{\zeta}' = \alpha(\hat{\zeta}_2 - \hat{\zeta}_1) + \hat{\zeta}_1$.

CN. The CN modules perform pixel value Clipping and Normalisation as a NN layer, as they can no longer be done during data loading as with pixel domain specifications.

To minimize over-approximation in verification, E should involve few non-linear operations, therefore all mentioned modules, except Clipping, are linear. While our implementation of E for conditional specifications encodes the two inputs being incorporated in a network layer and is reconstructed for every image batch, our E for the single input specifications takes the image batch as an input. Therefore, for the latter, we avoid the need to reconstruct E and reinitialise its bound propagation-wrapper per input batch. Avoiding this step improves efficiency as this step can be up to an order of magnitude slower than an interval or input propagation through the network.

3.2. Specification Design

While verification with the proposed framework is agnostic to any particular definition of the input perturbation set $\mathcal{B}_{p,\hat{\epsilon},\hat{M}}$, defining a Fourier domain input specification typically involves the following design choices:

- Fourier domain resolution. As mentioned earlier, encoding perturbations with frequencies lower than ω_f can require many FT coefficients. To encode some such perturbations with fewer coefficients, the frequency resolution can be increased as discussed in App.A.
- Fourier coefficients to perturb. This is dependent on the frequency spectrum of the intended perturbations and set using mask $\hat{M}$ for all perturbation models (2)-(5).
- Perturbation magnitudes $\hat{\epsilon}$ for Fourier amplitudes and

phases given the following observations:
The modularity of the Fourier coefficient components in that the amplitudes capture the style elements and the phases capture the structure and contours in the image. We use this to define two-input conditional specifications. *The magnitude of pixel space perturbations* to be verified against. To this end, we present two theorems to guide the setting of the DFT-domain epsilon given a desired pixel-domain epsilon in ℓ_∞, ℓ_2 norms. They are presented for $1d$ data for simplicity, and should be applied twice for $2d$ data. Their proofs are provided in App.B.

Theorem 1 *[$\mathcal{B}_{\infty,\hat{\epsilon},\hat{M}} \rightarrow \mathcal{B}_{\infty,\epsilon}$] The smallest pixel space ℓ_∞-ball $\mathcal{B}_{\infty,\epsilon}(0)$ enclosing the output set of IDFT for input set $\mathcal{B}_{\infty,\hat{\epsilon},\hat{M}}(0)$ has $\epsilon_\infty = d\hat{\epsilon}_\infty$, where d is the maximum number of DFT coefficients allowed to perturb by mask $\hat{M}$, i.e., $\max_{\hat{x}\in\mathcal{B}_{\infty,\frac{\epsilon_\infty}{d},\hat{M}}(0)} \|\bar{\mathcal{F}}^1(\hat{x})\|_\infty = \epsilon_\infty$.*

Theorem 2 *[$\mathcal{B}_{2,\hat{\epsilon},\hat{M}} \rightarrow \mathcal{B}_{2,\epsilon}$] The output of IDFT for input $\mathcal{B}_{2,\hat{\epsilon},\hat{M}}(0)$ is the pixel space ℓ_2-ball $\mathcal{B}_{2,\epsilon}(0) \in \mathbb{R}^D$ with $\epsilon_2 = \sqrt{D}\hat{\epsilon}_2$, i.e., $\{\bar{\mathcal{F}}^1(x) \mid x \in \mathcal{B}_{2,\frac{\epsilon_2}{\sqrt{D}},\hat{M}}(0)\} = \mathcal{B}_{2,\epsilon_2}(0)$.*

With both relations, the output of IDFT for $\mathcal{B}_{\hat{\epsilon},\hat{M}}(0)$ encloses perturbations with norm up to the corresponding pixel space ϵ and frequencies allowed by $\hat{M}$. The choice of relation could depend on the robustness guarantee desired and the adversarial attack to defend against, as discussed in the next section.

Once a specification is defined in terms of Fourier amplitudes and phases, it is soundly converted to a set in the Cartesian form as required by E. This conversion introduces some approximations and is discussed in App.B.

3.3. Capturing Frequency-domain Attacks.

The frequency domain offers great flexibility in the design of attacks. Fourier transform-based attacks alone range from frequency-band-contained attacks [45], frequency-band-mixup attacks [18] to domain-targeting attacks [16]. Given such diversity in attacks, it is desirable that the specifications are configurable enough to capture the same. The specification design choices, discussed above, cover the design space for most frequency-based attacks as discussed in Section 2 in [34]. For the systematic evaluation of our encoding and verification framework, we choose to use low-frequency specifications, i.e., sets spanning from zero frequency up to specified high frequencies, in our experiments. This is because compared to the high frequency attacks, [12, 26, 41] show that the low-frequency attacks are more transferable across networks and critical to defend against. This is since a) the typical pixel-domain adversarial training rarely samples from the low frequency regions to induce robustness against them, and b) the typical image processing such as compression may filter out and weaken the high-frequency attacks, not the low frequency ones.

3.4. Verification and Certified Training

Given the encoding network E and the input specifications defined in the space of DFT coefficients, the FDVP transforms to the standard transformation-parameters-based verification problem described in §2. The certified training for the specifications now involves inner maximization over the DFT coefficients, instead of the pixel values, i.e., $\min_\theta \mathbb{E}_{(x,y)\in(\mathcal{X},\mathcal{Y})}\left[\max_{\hat{\delta},\alpha} L\left(N_\theta \circ E(x)(\hat{\delta},\alpha), y\right)\right]$. As such, any differentiable bound-propagation framework that supports all operations in the network $N_\theta \circ E(x)$ can be used for its verification and certified training.

4. Experimental Evaluation

We evaluate our approach on the following criteria:

- the expressiveness and efficiency of the proposed DFT domain specifications over the existing ones (§4.1),
- the gains in NN verification from using the proposed specifications and encoding network over the pixel domain specifications (§4.2),
- the effectiveness of the proposed certified training in improving NN's verified accuracy (§4.3).

Following the standard network-dataset-verifier setups from existing works, we use CNN7 [28] and CNNDeep [8] networks, Traffic Signs Recognition (TSRD) and CIFAR10 datasets, and SoA verifiers AutoLiRPA [38] (incomplete) and $\alpha\beta$-CROWN [33] (complete) in our experiments.

4.1. Specification Efficacy

Specification Instances We use the DFT-domain specifications defined as per §3.2 to encode some realistic perturbation classes, and analyse representative samples enclosed by each of them in Fig 4.

Additive and multiplicative perturbations, such as those varying uniformly in pixels (brightness, contrast), independently for each pixel (whitenoise) or spatial intensity variations in between the two extremes. Such variations, shown in Fig 4a, are encoded using perturbation models (2) and (3). For changes uniform in all pixels, only the lowest frequency coefficient is perturbed. Perturbing higher frequency coefficients captures greater spatial variance up to whitenoise when all coefficients are perturbed.

Convolution-based perturbations, such as blur and sharpening achieved by convolution with Gaussian and Laplacian kernels. Given DFT's circular convolution property, perturbation model (4) is used to encode such variations resulting from pixel-space convolution. The coefficients to be perturbed depend on the FT of the kernels. If the DFT is monotonic w.r.t. the kernel parameter being varied to produce the perturbations, then the perturbation set is bounded by the FTs of the boundary kernels. Such an example is Gaussian blur, whose kernel is monotonic in the standard deviation parameter determining the extent of blur and shown in Fig 4b.

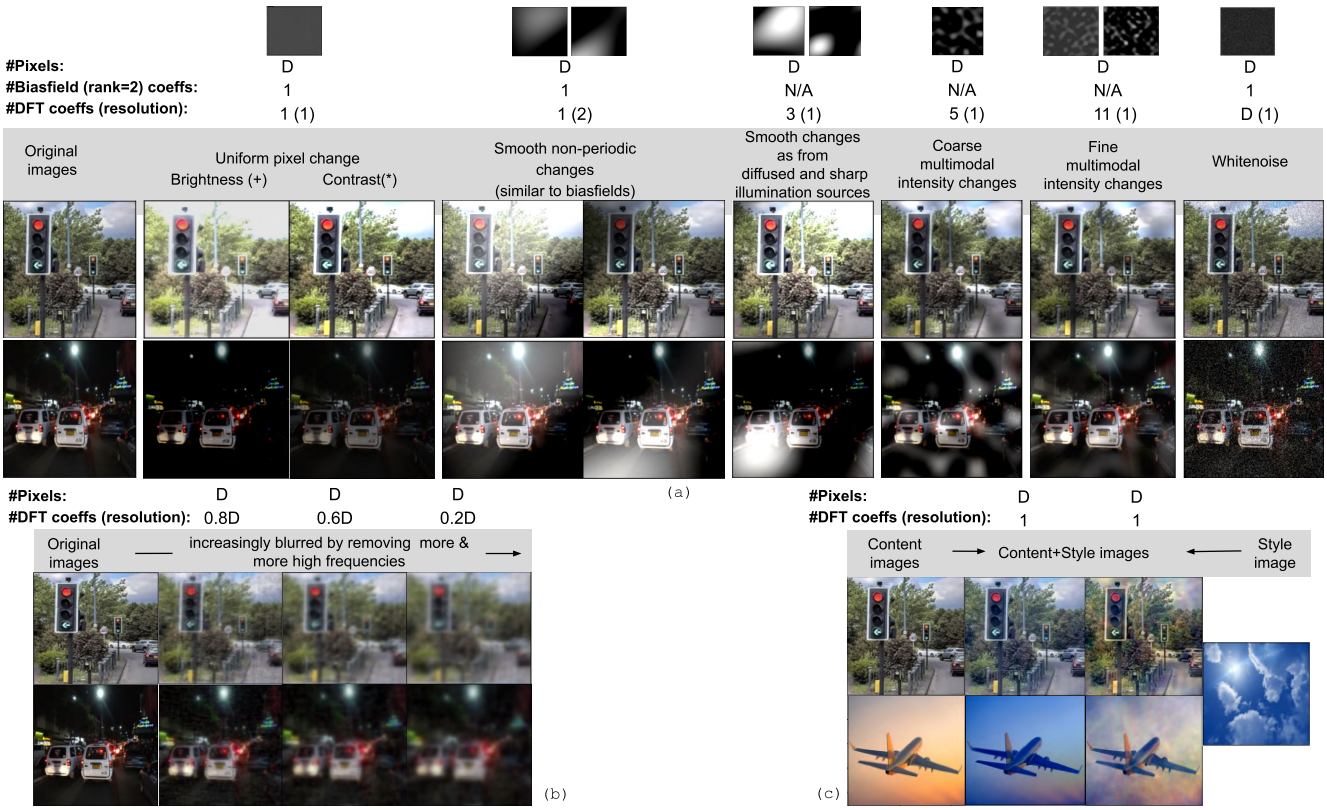


Figure 4. Representative perturbation samples from specifications in §4.1. a) shows additive (2) and multiplicative (3) perturbations from perturbing the specified number of DFT coefficients; b) shows blurred samples from a Gaussian kernel-based convolution specification (4); c) shows variations in only the style, not the content, of an image as per conditional specification (5) between it and another image.

Conditional domain change, such that the structural content of an image is preserved and only its visual style varies as in Fig 4c. For such changes, the specification is a segment between the Fourier amplitudes of an image pair from the original (X) and target (X') domains as in (5).

Specification Efficacy for Verification As mentioned in §2, fewer encoding variables allow using Forward BP and techniques such as input splitting [14] to obtain tighter NN bounds and thereby more precise verification outcomes. Therefore, we report in Fig 4 and compare the number of variables needed to encode different perturbations in the pixel, biasfield and DFT domains. For pixel-domain specifications, the number of encoding variables required is always the image dimensionality D. For blurring and style variations, the DFT coefficients required are always much fewer than D. For additive and multiplicative perturbations, the DFT and biasfield coefficients to encode smooth variations are much fewer than D. As the spatial variation in these perturbations increases to white noise, notice that:

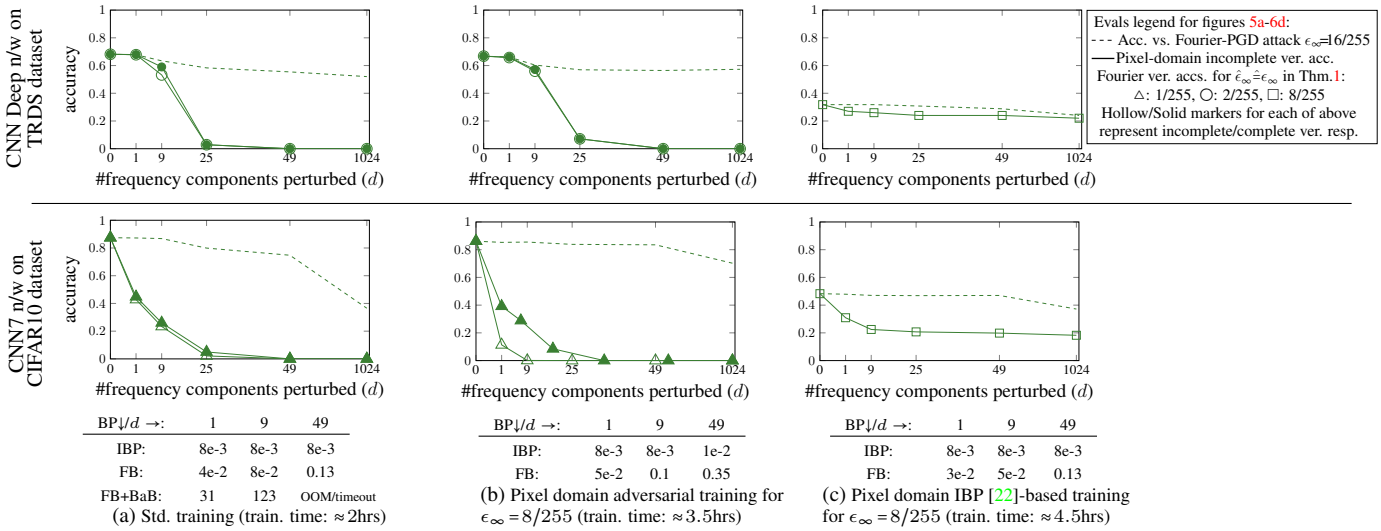
- With *biasfields*, all perturbations can no longer be encoded by the biasfield basis of finite rank. While some of them can still be encoded by increasing the rank, it would require constructing more higher-order spatial polynomials and adding them as NN layers to its encoding network E. In contrast, the DFT-based approach does not require any change in E's architecture for encoding the entire range of spatial variations.
- With *DFT*, the number of encoding coefficients also grow to D, showing that Fourier-space specifications have diminishing gains over pixel-space specifications as the frequency spectrum of the perturbations widens.

Summary. We validate that the perturbation models (2)-(5) implemented using E encode a wide range of realistic perturbations for a downstream network, and analyse the optimality of Fourier as a domain for specification design.

4.2. Verification for Fourier Specifications

We verify networks trained on two datasets with different pixel domain approaches, i.e., standard augmentation, adversarial augmentation and IBP-based certified training, for our additive[1] Fourier specifications (2). Their adversarial accuracy against Fourier-based PGD attack (the low-pass attack from [45]) and verified accuracies using complete and incomplete verifiers are reported in Fig 6. Verified accuracies for the networks are computed with using both Forward+Backward (FB) and Interval (IBP) bound propagation, with the the plotted accuracy being the highest ob-

[1]Experiments for conditional specifications feature in App.D.1.



BP↓/d →:	1	9	49
IBP:	8e-3	8e-3	8e-3
FB:	4e-2	8e-2	0.13
FB+BaB:	31	123	OOM/timeout

(a) Std. training (train. time: ≈2hrs)

BP↓/d →:	1	9	49
IBP:	8e-3	8e-3	1e-2
FB:	5e-2	0.1	0.35

(b) Pixel domain adversarial training for $\epsilon_\infty = 8/255$ (train. time: ≈3.5hrs)

BP↓/d →:	1	9	49
IBP:	8e-3	8e-3	8e-3
FB:	3e-2	5e-2	0.13

(c) Pixel domain IBP [22]-based training for $\epsilon_\infty = 8/255$ (train. time: ≈4.5hrs)

Figure 5. Adversarial (dashed line) and verified (solid line) accuracies of differently trained networks for additive specifications (2). The specifications allow a total of the x-axis specified number (d) of DFT coefficients including and around the central (zero frequency) coefficient to independently vary up to epsilon $\hat{\epsilon}_\infty$. The $\hat{\epsilon}_\infty$ is set corresponding to commonly used pixel-domain ϵ_∞ values ($\{1, 2, 8\}/255$) as per Thm.1. Given the high cost of complete verification (BaB), it is done only for non-certifiably trained networks. The three columns correspond to the three different approaches used to train the network. The three tables report the median verification time with different bound propagation approaches for each training approach. The details on networks, training and attack are described in App.C.

tained using both BP methods.

Verified Specifications. As motivated in §3.3, we verify against specifications that progressively allow perturbations of higher frequencies, starting with lowest-frequency uniform changes (when $d = 1$) up to the high-frequency white noise (when $d = 1024$ in Fig 6).

The main observations to be drawn from Fig 6 are:

- the non-trivial verified accuracies for the low-frequency specifications for all networks, even those that are not certifiably trained. Given that non-certifiably trained networks hardly register verified robustness against pixel-domain specifications, being able to verify them as robust for interpretable frequency-based specifications may be sufficient to validate their deployment for some safety-critical tasks.
- the closeness between the verified accuracies and adversarial accuracy (against the low-pass Fourier PGD attack [45]) for low-frequency specifications. The verified accuracy is an upper bound on the success of any attack within the input constraints enforced by the specification. While it is desirable to have the adversarial and verified accuracies close to each other, they typically have a large gap for pixel-space specifications and non-certifiably trained networks. Even in our results, as the input specifications become wider to include all frequencies, the gap between the two widens as propagating the broader input specifications adds in more over-approximation.

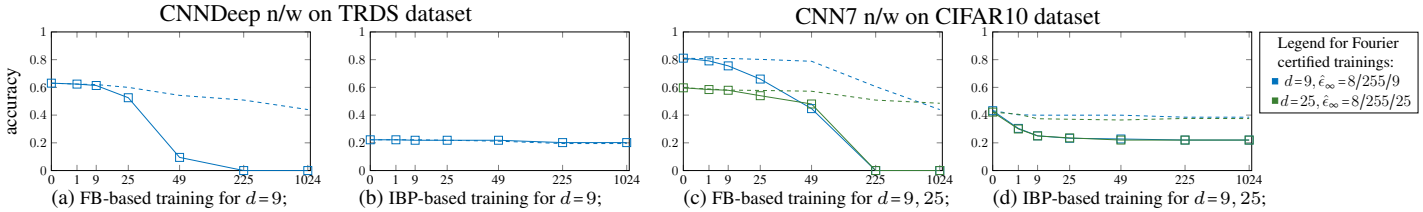
With regards to the effect of different training approaches, as expected, the verified accuracy stays consistent over more frequencies for the pixel space certifiably-trained network, but it is lower than the verified accuracies of the non-certifiably trained networks for the low-frequency specifications. It is due to the over-regularisation induced by the pixel space certified-training, a concern discussed in §4.3.

Computational expense. From the median verification times reported in the tables for each training approach, observe the moderate computational expense of FB-based incomplete verification for low specification dimensionality (d) for all networks, though it increases exponentially with increase in d. Additionally, the expense of BaB for complete verification is orders-of-magnitude higher and leads to out-of-memory or timeouts for bigger ds, ϵs and networks.

Counterexample analysis Counterexamples are instances of the specification that falsify the robustness property being verified. Table 1 shows some counterexamples for our specifications as they grow to include higher frequencies. Following [27], we also use the Frechet Inception Distance (FID) [15] on these counterexamples as their stealthiness indicator. We report these for a high epsilon $\epsilon = 32/255$ to make the perturbations in the counterexamples perceptible, and enhance the FID between them and the clean images. Notice from Table 1 that as the highest

ϵ \ d	1	9	25	49	Pixel
$\frac{32}{255}$	16.7	81.7	109.82	129.2	178.6

Table 1. Representative counterexamples for specifications that progressively include higher frequencies as d increases, and the FID between these counterexamples and the original images.



(a) FB-based training for $d=9$; (training time: $\approx$3hrs)
(b) IBP-based training for $d=9$; (training time: $\approx$1hr)
(c) FB-based training for $d=9, 25$; (training time: $\approx$10hrs)
(d) IBP-based training for $d=9, 25$; (training time: $\approx$5hrs)

Figure 6. Adversarial (dashed) and verified (solid) accuracies of networks certifiably-trained for Fourier additive specifications (2) of Fourier-domain $\hat{\epsilon}_\infty$ corresponding to pixel-space $\epsilon_\infty = 8/255$ as per Thm 1. The d in the captions denotes the number of DFT coefficients allowed to perturb during training. The verified specifications and plots are same as for Fig 6, refer to its caption and legend for details.

frequency in the specification increases, the perturbations change from uniform to smooth multimodal intensity change to pixelated, and the FID increases indicating that the counterexamples become more out-of-distribution from the original images.

Summary. We validate that the proposed approach verifies networks as robust against low-frequency perturbation sets and adversarial attacks, thus offering a low-dimensional verification guarantee when one cannot be obtained for the high-dimensional pixel space sets.

4.3. Certified Training for Fourier Specifications

We prepend E corresponding to additive perturbation model (2) to our networks and train them for robustness against the low frequency Fourier-domain specifications by using the standard robust cross-entropy loss, and the FB and IBP bound propagations. The training setup is in line with existing certified training schemes and detailed in App.C. From the verified accuracies of the Fourier domain-based certifiably-trained networks in Fig 6c-6d, observe:

- Their higher verified accuracies for specifications of higher magnitudes and including more high frequencies than their non-certifiably trained counterparts in Fig 5a-5b, thus validating our certified training implementation,
- The high verified and standard accuracy of the FB-trained networks in Fig 6c. This is expected since, given the prepended E, the FB-based bound propagation trains and evaluates a network for perturbation sets which ideally, as in with a complete verifier, should only include perturbations composed of frequencies allowed by the specification. These sets for low-frequency specifications should be much smaller than a set of perturbations allowed by the ℓ_∞-norm pixel domain specifications where the pixels can vary independently. As such, training for these smaller perturbation sets results in certified robustness against only these specific perturbations and does not extend to the pixel-domain whitenoise; however, the standard network accuracy stays high due to less over-regularisation in training. Note that it requires much higher training time than the other approaches,
- The similarity in the robustness characteristics of the IBP-trained networks in Fig 6d to that of the pixel domain-based certifiably trained network in Fig 5c. This is expected since IBP, due to its per-layer concretisation of bounds, does not propagate the interdependence among nodes as enforced by the prepended E. Thus, irrespective of the perturbation mask and dimensionality, this training is similar to training for an ℓ_∞-norm pixel domain set that encompasses the allowed frequency-domain perturbations. As such, it reports much lower standard accuracy than the FB-trained networks, but a non-trivial certified accuracy against the pixel-space whitenoise.

Summary. The high verified accuracies of the certifiably-trained augmented networks $E \circ N_\theta$ using FB bounds validate that the proposed certified training increases network robustness against low-frequency perturbations and attacks, with a decent standard-vs-verified accuracy trade-off. Although more expensive than IBP, FB is reasonably efficient to compute for low-dimensional specifications, supporting the feasibility of our training proposal.

5. Conclusions

Given the need for deterministic robustness assessment of NNs for autonomous deployments and the ubiquity of Fourier-based attacks, this work proposes configurable input specifications in the Fourier domain, and a framework to enable their verification and certified training using SoA tools. We show that for our low-frequency specifications, networks can be verified as non-trivially robust without certified training, and that Fourier-domain based certified training can further enhance this verified robustness without drastically trading off standard accuracy. Future work as motivated by current limitations include making the encoding network directly support Fourier specifications in Polar form to make them absolutely precise and making the perturbations composable.

Acknowledgements

Alessandro De Palma was supported by the "SAIF" project, funded by the "France 2030" government investment plan managed by the French National Research Agency, under the reference ANR-23-PEIA-0006. Alessio Lomuscio was supported by a Royal Academy of Engineering Chair in Emerging Technologies.

References

[1] Motasem Alfarra, Adel Bibi, Naeemullah Khan, Philip HS Torr, and Bernard Ghanem. Deformrs: Certifying input deformations with randomized smoothing. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2022. 2

[2] Pranjal Awasthi, Himanshu Jain, Ankit Singh Rawat, and Aravindan Vijayaraghavan. Adversarial robustness via robust low rank representations. *Advances in Neural Information Processing Systems*, 2020. 2

[3] Pranjal Awasthi, George Yu, Chun-Sung Ferng, Andrew Tomkins, and Da-Cheng Juan. Adversarial robustness across representation spaces. *CoRR*, abs/2012.00802, 2020. 1, 2

[4] M. Balunovic, M. Baader, G. Singh, T. Gehr, and M. Vechev. Certifying geometric robustness of neural networks. In *Advances in Neural Information Processing Systems (NeurIPS19)*. 2019. 2

[5] R.N. Bracewell. *The Fourier Transform and its Applications*. McGraw-Hill Kogakusha, Ltd., second edition, 1978. 3

[6] Rudy Bunel, Ilker Turkaslan, Philip HS Torr, Pushmeet Kohli, and M Pawan Kumar. A unified view of piecewise linear neural network verification. *Neural Information Processing Systems*, 2018. 3

[7] Muxi Chen, Zhijian Xu, Ailing Zeng, and Qiang Xu. Fraug: Frequency domain augmentation for time series forecasting, 2023. 2

[8] Alessandro De Palma, Harkirat Singh Behl, Rudy Bunel, Philip H. S. Torr, and M. Pawan Kumar. Scaling the convex barrier with active sets. *International Conference on Learning Representations*, 2021. 3, 5

[9] Alessandro De Palma, Rudy Bunel, Krishnamurthy Dvijotham, M Pawan Kumar, Robert Stanforth, and Alessio Lomuscio. Expressive losses for verified robustness via convex combinations. In *International Conference on Learning Representations*, 2024. 3

[10] S. Dutta, S. Jha, S. Sankaranarayanan, and A. Tiwari. Output range analysis for deep feedforward neural networks. In *NASA Formal Methods*, pages 121–138, Cham, 2018. Springer International Publishing. 3

[11] S. Gowal, K. Dvijotham, R. Stanforth, R. Bunel, C. Qin, J. Uesato, R. Arandjelovic, T. Mann, and P. Kohli. Scalable verified training for provably robust image classification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (IEEE/CVF19)*, 2019. 3

[12] Chuan Guo, Jared S. Frank, and Kilian Q. Weinberger. Low frequency adversarial perturbation. In *Proceedings of the Thirty-Fifth Conference on Uncertainty in Artificial Intelligence, UAI 2019, Tel Aviv, Israel, July 22-25, 2019*, volume 115 of *Proceedings of Machine Learning Research*, pages 1127–1137. AUAI Press, 2019. 1, 5

[13] H. Hanspal and A. Lomuscio. Efficient verification of neural networks against lvm-based specifications. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR23)*. IEEE, 2023. 2

[14] P. Henriksen, K. Hammernik, D. Rueckert, and A. Lomuscio. Bias field robustness verification of large neural image classifiers. In *Proceedings of the 32nd British Machine Vision Conference (BMVC21)*. BMVA Press, 2021. 2, 3, 6

[15] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. 7

[16] Jiaxing Huang, Dayan Guan, Aoran Xiao, and Shijian Lu. RDA: robust domain adaptation via fourier adversarial attacking. *CoRR*, abs/2106.02874, 2021. 1, 2, 5

[17] G. Katz, C. Barrett, D. Dill, K. Julian, and M. Kochenderfer. Reluplex: An efficient SMT solver for verifying deep neural networks. In *Proceedings of the 29th International Conference on Computer Aided Verification (CAV17)*, Lecture Notes in Computer Science. Springer, 2017. 2, 3

[18] Xiu-Chuan Li, Xu-Yao Zhang, Fei Yin, and Cheng-Lin Liu. F-mixup: Attack cnns from fourier perspective. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 541–548, 2021. 1, 2, 5

[19] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *International Conference on Learning Representations*, 2018. 3

[20] M. Mirman, A. Hägele, P. Bielik, T. Gehr, and M. Vechev. Robustness certification with generative models. In *Proceedings of the 42nd ACM SIGPLAN International Conference on Programming Language Design and Implementation*, 2021. 2

[21] Jeet Mohapatra, Tsui-Wei Weng, Pin-Yu Chen, Sijia Liu, and Luca Daniel. Towards verifying robustness of neural networks against a family of semantic perturbations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020. 2, 3

[22] M.N. Müller, F. Eckert, M. Fischer, and M.T. Vechev. Certified training: Small boxes are all you need. In *Proceedings of the 11th International Conference on Learning Representations (ICLR23)*, 2023. 3, 7

[23] Yanqi Qiao, Dazhuang Liu, Rui Wang, and Kaitai Liang. Low-frequency black-box backdoor attack via evolutionary algorithm, 2024. 2

[24] A. Raghunathan, J. Steinhardt, and P. Liang. Semidefinite relaxations for certifying robustness to adversarial examples. *arXiv preprint arXiv:1811.01057*, 2018. 3

[25] Anian Ruoss, Maximilian Baader, Mislav Balunović, and Martin Vechev. Efficient certification of spatial robustness. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, 2021. 2

[26] Yash Sharma, Gavin Weiguang Ding, and Marcus A. Brubaker. On the effectiveness of low frequency perturbations. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, page 3389–3396. AAAI Press, 2019. 1, 5

[27] D. Shelepneva and K. Arkhipenko. Realistic adversarial attacks on object detectors using generative models. *Journal of Mathematical Sciences*, 285(2):245–254, Oct 2024. 7

[28] Z. Shi, Y. Wang, H. Zhang, J. Yi, and C. Hsieh. Fast certified robust training via better initialization and shorter

warmup. In *Advances in Neural Information Processing Systems (NeurIPS21)*, pages 18335–18349, 2021. 3, 5

[29] Gagandeep Singh, Timon Gehr, Matthew Mirman, Markus Püschel, and Martin Vechev. Fast and effective robustness certification. *Neural Information Processing Systems*, 2018. 3

[30] Yusuke Tsuzuku and Issei Sato. On the structural sensitivity of deep convolutional networks to the directions of fourier basis functions. *CoRR*, abs/1809.04098, 2018. 1, 2

[31] An Wang, Mobarakol Islam, Mengya Xu, and Hongliang Ren. Curriculum-based augmented fourier domain adaptation for robust medical image segmentation, 2023. 2

[32] S. Wang, K. Pei, J. Whitehouse, J. Yang, and S. Jana. Formal security analysis of neural networks using symbolic intervals. In *Proceedings of the 27th USENIX Security Symposium (USENIX18)*, 2018. 3

[33] S. Wang, H. Zhang, K. Xu, X. Lin, S. Jana, C. Hsieh, and J. Kolter. Beta-crown: Efficient bound propagation with per-neuron split constraints for complete and incomplete neural network verification. *arXiv preprint arXiv:2103.06624*, 2021. 3, 5

[34] Tong Wang, Yuan Yao, Feng Xu, Shengwei An, Hanghang Tong, and Ting Wang. An invisible black-box backdoor attack through frequency domain. In *Computer Vision-ECCV 2022*. Springer Nature Switzerland, 2022. 2, 5

[35] Zerui Wen. Fourier attack – a more efficient adversarial attack method. In *Proceedings of the 6th International Conference on Control Engineering and Artificial Intelligence*, 2022. 1, 2

[36] E. Wong, F. Schmidt, and Z. Kolter. Wasserstein adversarial examples via projected Sinkhorn iterations. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 6808–6817. PMLR, 09–15 Jun 2019. 2

[37] E. Wong, F. Schmidt, J. Metzen, and J. Kolter. Scaling provable adversarial defenses. In *Proceedings of the 32nd Conference on Neural Information Processing Systems (NeurIPS18)*, 2018. 3

[38] Kaidi Xu, Zhouxing Shi, Huan Zhang, Yihan Wang, Kai-Wei Chang, Minlie Huang, Bhavya Kailkhura, Xue Lin, and Cho-Jui Hsieh. Automatic perturbation analysis for scalable certified robustness and beyond. *Advances in Neural Information Processing Systems*, 33, 2020. 5

[39] Kaidi Xu, Huan Zhang, Shiqi Wang, Yihan Wang, Suman Jana, Xue Lin, and Cho-Jui Hsieh. Fast and complete: Enabling complete neural network verification with rapid and shimassively parallel incomplete verifiers. In *International Conference on Learning Representations*, 2021. 3

[40] Qinwei Xu, Ruipeng Zhang, Ziqing Fan, Yanfeng Wang, Yi-Yan Wu, and Ya Zhang. Fourier-based augmentation with applications to domain generalization. *Pattern Recognition*, 139:109474, 2023. 2

[41] Dong Yin, Raphael Lopes, Jonathon Shlens, Ekin Cubuk, and Justin Gilmer. A fourier perspective on model robustness in computer vision. 06 2019. 2, 5

[42] H. Zhang, H. Chen, C. Xiao, S. Gowal, R. Stanforth, B. Li, D. Boning, and C. Hsieh. Towards stable and efficient training of verifiably robust neural networks. *arXiv preprint arXiv:1906.06316*, 2019. 3

[43] Huan Zhang, Hongge Chen, Chaowei Xiao, Sven Gowal, Robert Stanforth, Bo Li, Duane S. Boning, and Cho-Jui Hsieh. Towards stable and efficient training of verifiably robust neural networks. In *International Conference on Learning Representations*, 2020. 3

[44] H. Zhang, P. Zhang, and C. Hsieh. Recurjac: An efficient recursive algorithm for bounding jacobian matrix of neural networks and its applications. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):5757–5764, Jul. 2019. 3

[45] Liangqi Zhang, Yihao Luo, Haibo Shen, and Tianjiang Wang. A fourier perspective of feature extraction and adversarial robustness. In Kate Larson, editor, *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, IJCAI-24*, pages 1715–1723. International Joint Conferences on Artificial Intelligence Organization, 8 2024. Main Track. 1, 2, 5, 6, 7