Perturbing the Phase: Analyzing Adversarial Robustness of Complex-Valued Neural Networks

Florian Eilers
Department of Computer Science
University of Münster
Münster, Germany

Christof Duhme
Department of Computer Science
University of Münster
Münster, Germany

Xiaoyi Jiang
Department of Computer Science
University of Münster
Münster, Germany

***Abstract*—Complex-valued neural networks (CVNNs) are rising in popularity for all kinds of applications. To safely use CVNNs in practice, analyzing their robustness against outliers is crucial. One well known technique to understand the behavior of deep neural networks is to investigate their behavior under adversarial attacks, which can be seen as worst case minimal perturbations. We design Phase Attacks, a kind of attack specifically targeting the phase information of complex-valued inputs. Additionally, we derive complex-valued versions of commonly used adversarial attacks. We show that in some scenarios CVNNs are more robust than RVNNs and that both are very susceptible to phase changes with the Phase Attacks decreasing the model performance more, than equally strong regular attacks, which can attack both phase and magnitude.**

I. Introduction

While most deep learning architectures are still real-valued neural networks (RVNNs), complex-valued neural networks (CVNNs) have lately risen in popularity. Their natural ability to deal with complex-valued inputs make them feasible for a number of applications that deal with complex-valued images including safety critical applications such as MR imaging [1], [2], autonomous driving [3] and many more [4]–[6].

Deep neural networks have been shown to be vulnerable to drastic output changes upon small input perturbations. The robustness of models against small distribution changes can be investigated by adversarial attacks. They can be understood as a ”worst case” minimal perturbation that maximizes the change of the output.

While many works have investigated how RVNNs react to adversarial attacks, no such investigation has been made for CVNNs. Besides introducing how to calculate adversarial attacks in the complex domain, we present a comparative study analyzing the robustness of RVNNs and CVNNs. To this end, we present a novel attack - the Phase Attack, that specifically targets the phase of a complex-valued input. The phase information is often neglected in visualization and is hard to understand, thus perturbations to it are especially prone to be unnoticed. To further evaluate the impact of phase perturbation, we also design its counterpart, the Magnitude Attack, which only targets the magnitude but leaves the phase unchanged. We show, that Phase Attacks are the most effective attacks, when compared to regular attacks and Magnitude attacks, showing the susceptibility of both CVNNs and RVNNs to phase perturbation, when dealing with complex valued inputs.

To summarize, we contribute:

- A new kind of attack: Phase Attacks, that are specifically suited to attack complex-valued inputs in real-world applications and analysis of its effectiveness.
- An comparative analysis on the adversarial robustness between CVNNs and RVNNs.
- A toolbox in PyTorch to apply all presented approaches to the complex domain. The code can be found at https://zivgitlab.uni-muenster.de/ag-pria/cv-attacks-phase-attacks.

II. Related Work

A. Adversarial Attacks

Since the definition of ”adversarial attacks” [7] the effect of minor perturbations fooling deep neural networks has been analyzed extensively [8]–[14]. More recently, deep neural networks were utilized to craft more powerful adversarial attacks against other models [15]. Additionally, defense algorithms have been proposed as counter measures against adversarial attacks [16]–[18], however many of them can be circumvented [19]. Adversarial training describes the counter measure, where adversarial attacks are applied during training and has been shown very effective [9], [10], [20]. Besides these practical works, many works have analyzed the robustness of deep neural networks and the effects, that lead to adversarial vulnerability [21], [22].

B. Complex-Valued Neural Networks

Since the early days of deep learning, CVNNs have been a point of interest [23]–[26]. More recently, deep CVNNs have been introduced with basic building blocks [27], as well as advances in model architectures [28]–[30], theoretical understanding [31], [32] and hardware optimization [33], [34]. They have lately risen in popularity for a multitude of applications [32], [35]–[38], including safety critical applications, such as MRI image processing [1], [39], [40] or autonomous driving [3]. The robustness to noise of analog CVNNs has been investigated in the context of optical neural networks [41].

Zhou et al. [42] have shown, that the phase information of the Fourier transform of adversarial attacks crafted for real-valued images is pivotal. They however do not use this

knowledge to craft an attack that focuses on the phase and do only investigate the effect of the phase of the Fourier transform of natural images processed by RVNNs, rather than also incorporating complex-valued images, which offer a phase information in image space.

In a theoretical work, Neaccsu et al. [43] present a Lipschitz bound for robustness against adversarial attacks of CVNNs and derive a shallow model architecture as well as a training routine to satisfy this Lipschitz bound, however their approach is based on nonnegative neural networks, which are unsuitable for deep neural networks. They do not compare the robustness of RVNNs and CVNNs.

Lastly, the closest related work to our work is a work by Yeats et al. [44]. They investigate how gradient regularization has different effects on adversarial robustness of CVNNs and RVNNs and conclude, that gradient regularization is more effective in CVNNs than in RVNNs. However, they only investigate this phenomenon on real-valued image datasets. We are interested in the setting, where the input data is complex-valued, since that is a primary use case for CVNNs.

To the best of our knowledge, no study has so far compared the adversarial robustness of RVNNs and CVNNs for complex-valued inputs or investigated the impact of phase perturbations to complex-valued inputs.

III. Gradient-Based Adversarial Attacks

An adversarial attack can loosely be defined as an imperceivable perturbation to input data of a deep learning model that significantly alters the output of the model. While adversarial attacks are mostly studied for images, the general framework is applicable to any input vector. We thus use the terms input (vector), which has entries, instead of speaking of images and pixels specifically.

Adversarial attacks can formally be formulated as a restricted optimization problem. Let f be some trained (real-valued) deep learning model, $X \in \mathbb{R}^n$ its input with label Y and some objective function l. Then the adversarial input Z^* is defined as:

$$Z^* = \text{argmax}_{Z \in \mathbb{R}^n} \, l(f(Z), Y) \quad \text{s.t.} \, ||Z - X||_\infty < \epsilon \tag{1}$$

The bound $\epsilon > 0$ is a small real number that is tuned depending on the dataset at hand. A greater value makes the attack stronger, but also leads to a more perceivable perturbation.

To solve this restricted optimization problem efficiently, gradient based optimization algorithms can be utilized. The Fast Gradient Sign Method (FGSM) [8] makes use of a single gradient step. The attacked input Z_{FGSM} is then defined as:

$$Z_{\text{FGSM}} = X + \epsilon \, \text{sign} \left(\nabla l(f(X)) \right) \tag{2}$$

where sign describes the point wise sign of the gradient vector. This basic single step approach can now be extended by, instead of starting with X, starting at a random position in the search space $B_{\epsilon,X} = \{Z \in \mathbb{R}^n | \; ||Z - X||_\infty < \epsilon\}$. This approach is called Faster FGSM (FFGSM) [10], the attacked input Z_{FFGSM} is then defined as:

$$Z_{\text{FFGSM}} = \mathcal{P}_{B_{\epsilon,X}} \left(X + U + \epsilon \, \text{sign} \left(\nabla l(f(X + U)) \right) \right) \tag{3}$$

where $U \in \mathbb{R}^n$ denotes a random vector from the uniform distribution $\mathcal{U}(B_\epsilon(0))$ and $\mathcal{P}_{B_{\epsilon,X}}$ denotes the projection onto $B_{\epsilon,X}$, ensuring the ϵ-bound.

This can be further extended by not just going a single gradient step, but iteratively refining the attacked input. This was first proposed by Kurakin et al. [45] (Basic Iterative Method) and was later extended to the Projected Gradient Descent by including a random starting position [9]. To ensure consistency in our notation, we call it Iterative FGSM (IFGSM). For some $0 < \alpha < \epsilon$ (often $\alpha = \frac{\epsilon}{4}$) and some maximum step m, the attacked input is defined as $Z_{\text{IFGSM}} = Z_m$ and then:

$$Z_0 = X + U \tag{4}$$

$$Z_{t+1} = \mathcal{P}_{B_{\epsilon,Z_t}} \left(Z_t + \alpha \, \text{sign} \left(\nabla l(f(Z_t)) \right) \right) \tag{5}$$

Lastly, a momentum term can be added to the gradient calculation. This leads to Momentum IFGSM (MIFGSM) [11] and is (with a momentum controlling parameter $0 \leq \beta \leq 1$) again defined as $Z_{\text{MIFGSM}} = Z_m$ with:

$$Z_0 = X + U; \; M_0 = 0 \tag{6}$$

$$M_t = M_{t-1} + \beta \, \text{sign} \left(\nabla l(f(Z_t)) \right) \tag{7}$$

$$Z_{t+1} = \mathcal{P}_{B_{\epsilon,Z_t}} \left(Z_t + \alpha \, \text{sign} \left(\nabla l(f(Z_t)) + M_t \right) \right) \tag{8}$$

Many variants and improvements to these four base methods have been proposed, such as using Nesterov Momentum [12], Variance Tuning [13], Input Diversity [14] and many more [46]. Since these are all specific improvements to the base methods above, we focus on the four base types to compare robustness between RVNNs and CVNNs.

IV. Complex-Valued Gradient-Based Attacks

In this section, we describe how to design complex-valued adversarial attacks. For the adaption we can rely on Wirtinger calculus [47], a well known tool for optimization of real-valued functions (such as loss functions) with complex-valued inputs. With $\mathcal{R}(z), \mathcal{I}(z)$ being the real- and imaginary parts of $z \in \mathbb{C}$ and i the complex unit, the Wirtinger derivatives for a function $f : \mathbb{C} \to \mathbb{K}$ with $\mathbb{K} = \mathbb{C}$ or $\mathbb{R}$ are defined as:

$$\frac{\delta f}{\delta z} := \frac{1}{2} \left(\frac{\delta f}{\delta \mathcal{R}(z)} - i \frac{\delta f}{\delta \mathcal{I}(z)} \right) \tag{9}$$

$$\frac{\delta f}{\delta \overline{z}} := \frac{1}{2} \left(\frac{\delta f}{\delta \mathcal{R}(z)} + i \frac{\delta f}{\delta \mathcal{I}(z)} \right) \tag{10}$$

The Wirtinger derivative w.r.t. the complex conjugate $\frac{\delta f}{\delta \overline{z}}$ points in the direction of the strongest ascent and the magnitude equals the amount of change in that direction [24], just like the gradient in the real domain. Thus we can use it to solve the optimization problem posed for adversarial attacks in the complex domain. Let $X \in \mathbb{C}^n$ be a complex-valued input with label Y to some (real-valued or complex-valued) model

f with objective function l and $\epsilon > 0$ the perturbation bound. The complex-valued version of (1) then states:

$$Z^* = \text{argmax}_{Z \in \mathbb{C}^n} \, l(f(Z), Y) \quad \text{s.t.} \, ||Z - X||_\infty < \epsilon \tag{11}$$

where the complex infinity norm is defined as:

$$||z||_\infty := \max_{1 \leq j \leq n} (|z_j|) \tag{12}$$

We now briefly discuss how to adapt all presented methods from Section III using Wirtinger derivatives. Since MIFGSM is the most advanced version of the previously introduced methods, we will define $\mathbb{C}$MIFGSM in detail, all other versions can then be achieved by simplification.

To this end, let $U_\mathbb{C}$ be a random input of size of X drawn from the complex uniform distribution on the ϵ-ball around 0: $\mathcal{U}_\mathbb{C}(B_{\epsilon,0})$. Additionally, let $0 \leq \beta \leq 1$ and $0 < \alpha < \epsilon$. Then, for some max step size m, $\mathbb{C}$MIFGSM is calculated as $Z^* = Z_m$ by:

$$Z_0 = X + U_\mathbb{C}; \;\; M_0 = 0 \tag{13}$$

$$M_t = M_{t-1} + \beta \; \text{sign} \left(\nabla_{\overline{z}} l(f(Z_t)) \right) \tag{14}$$

$$Z_{t+1} = \mathcal{P}_{B_{\epsilon,Z_t}} \left(Z_t + \alpha \; \text{sign} \left(\nabla_{\overline{z}} l(f(Z_t)) + M_t \right) \right) \tag{15}$$

With ϕ_Z the phase of $Z \in \mathbb{C}$ its sign is defined as:

$$\text{sign}(Z) = \text{sign}(|Z| \exp(i\phi_Z)) = \exp(i\phi_Z) \tag{16}$$

Additionally, $\nabla_{\overline{z}}$ denotes the Wirtinger gradient with partial derivatives defined as in (10) and $\mathcal{P}_{B_{\epsilon,Z_t}}$ denotes the projection onto B_{ϵ,Z_t}). From (13)-(15) we can define $\mathbb{C}$IFGSM, by setting $\beta = 0$, $\mathbb{C}$FFGSM by also setting $\alpha = \epsilon$ and $m = 1$ and finally $\mathbb{C}$FGSM by setting $U_\mathbb{C} = 0$.

V. Phase Attacks

The attacks presented in Section IV are straightforward extension from the real domain, however they do not consider the unique nature of the complex domain. When handling complex data in practical applications, it is common practice to only visualize the magnitude information, since the phase information is normally very noisy and hard to understand. Even if the phase information is visualized, perturbations within the phase are hard to detect due to the heavy noise, especially in regions of low signal magnitude. The basic idea of our proposed Phase Attacks is to specifically target the phase of the complex-valued input and leave the magnitude unperturbed to specifically analyze, how sensitive a deep neural network is to phase perturbations.

To define the Phase Attack we keep the ϵ-restraint from Section IV but additionally add the constraint, that we do not perturb the magnitude of the input. Let $X \in \mathbb{C}^n$ be an input to be perturbed with label Y, f a (real- or complex-valued) model to predict Y from X and l a suitable loss function. We then have to solve:

$$Z^* = \text{argmax}_{Z \in \mathbb{C}^n} \, l(f(Z), Y) \quad \text{s.t.} \, ||Z - X||_\infty < \epsilon \text{ and } |Z| \underset{p.w.}{=} |X| \tag{17}$$

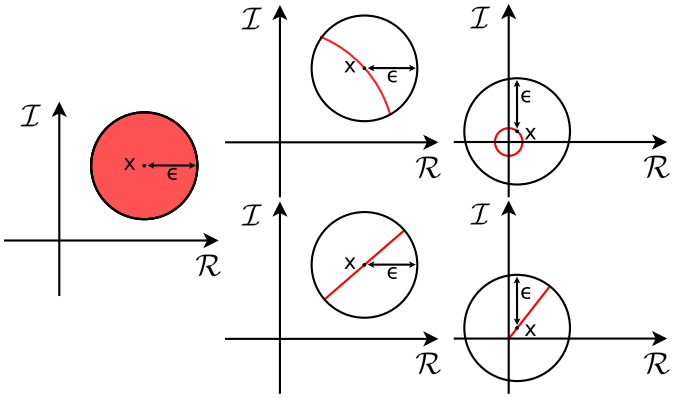

Fig. 1. Visualization of search spaces for adversarial attack optimization problems. Black circle shows epsilon ball around value X, search space in red. Left: Classical adversarial attack (17), center top/bottom: Phase Attack/Magnitude Attack for X far from zero, right top/bottom: Phase Attack/Magnitude Attack for X close to zero ((18)/(27)).

This is equivalent to solving the following:

$$Z^* = |X| \exp(i(\phi_X + \phi_Z^*)) \text{ with} \tag{18}$$
$$\phi_Z^* = \text{argmax}_{\phi_Z \in [-\pi,\pi)} \, l(f(|X| \cdot \exp(i(\phi_X + \phi_Z))), Y)$$
$$\text{s.t.} \;\; 2|X| \underset{p.w.}{\leq} \epsilon \text{ or } |\phi_X - \phi_Z| \underset{p.w.}{<} 2 \arcsin \left(\frac{\epsilon}{2|X|} \right)$$

Note that we apply both restrictions point wise to the input and the logical 'or' also needs to be understood pointwise. In other words: For any entry, one of the two pointwise restrictions needs to hold. The first restriction means that we can freely perturb the phase of entries with magnitude close enough to 0, while the second restriction shows that for entries with increasing magnitude the allowed phase perturbations decrease. We have visualized this in Fig. 1. A proof for the equivalence of the two optimization problems in (17) and (18) can be found in the appendix[1].

A. Optimization Algorithms for Phase Attacks

We present an optimization algorithm to solve the optimization problem in (18). Our approach is inspired by the existing methods presented in Section IV. Just like $\mathbb{C}$FGSM, the general idea of our optimization algorithm is to determine an optimal direction of change for every entry (which is guided by the gradient) and then make a maximum step into that direction.

We describe the one step attack in detail, which can then be extended to a multi step attack similarly to how $\mathbb{C}$FGSM can be extended to $\mathbb{C}$MIFGSM by starting in a random position in the search space, applying the steps multiple times and replacing the gradient by a momentum enhanced gradient. We call the respective concrete implementations of Phase Attacks PFGSM, PFFGSM, PIFGSM and PMIFGSM.

Since the formulation in (18) consists of two restrictions, we have to differentiate which restriction is limiting the perturbation of a specific entry. As highlighted before, if

[1]For the appendix, please refer to the code (link above).

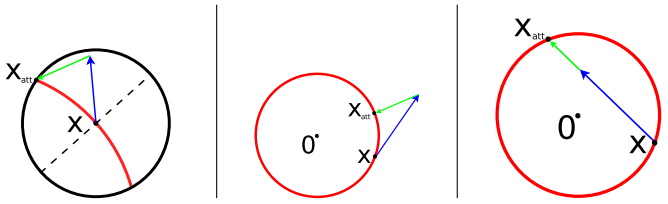

Fig. 2. Visualization of optimization steps for Phase Attack optimization problems. Search space in red (see Fig. 1), Wirtinger gradient in blue. Left: restricted phase perturbation (25), center: freely perturbed phase with outward pointing gradient (21), right: freely perturbed phase with inward pointing gradient (22).

$2|X| \leq \epsilon$ holds in an entry, we can freely perturb the phase, if this does not hold, we have to restrict the phase perturbation to $2\arcsin(\epsilon/(2|X|))$. In the first case, we differentiate further between two scenarios: The Wirtinger derivative of that entry points into or out of the search space (which is just a sphere in this case). All three scenarios are highlighted below and visualized in Fig. 2.

1) Optimization for Unrestricted Phase Perturbations: We assume, we can freely perturb the phase (i.e., the condition $2|X| \leq \epsilon$ holds for an entry). Thus, we want to perturb the phase of an entry of an input to model f. Let $X \in \mathbb{C}$ be the corresponding value of that entry and $\frac{\delta f}{\delta \overline{x}} \in \mathbb{C}$ its Wirtinger derivative w.r.t. the complex conjugate. We now have to differentiate between two scenarios: When looking at the circle with magnitude $|X|$, then $\frac{\delta f}{\delta \overline{x}}$ either points outwards that circle from X or inwards (see Fig. 2). Precisely, we have to distinguish between:

$$\text{Pointing outwards: } \mathcal{R}\left(\frac{\delta f}{\delta \overline{x}}\overline{X}\right) \geq 0 \tag{19}$$

$$\text{Pointing inwards: } \mathcal{R}\left(\frac{\delta f}{\delta \overline{x}}\overline{X}\right) < 0 \tag{20}$$

Note that $\mathcal{R}(a\bar{b}) = \langle(\mathcal{R}(a), \mathcal{I}(a)), (\mathcal{R}(b), \mathcal{I}(b))\rangle_{\mathbb{R}^2}$ holds for $a, b \in \mathbb{C}$, which leads to an easily understandable interpretation of the inequality in (19). The formulation above is analogous to a formulation we need later for the optimization in the restricted case.

Optimization in case of outward pointing derivative
In this case, we use the directional information (i.e., the sign) and the strength information (i.e., the magnitude) of the gradient to generate an attack. A regular gradient descent step would lead to $X + \frac{\delta f}{\delta \overline{x}}$, however $|X + \frac{\delta f}{\delta \overline{x}}| > |X|$, thus we need a projection onto our search space, the circle of magnitude $|X|$. To summarize, the Phase Attack in the unrestricted case with an outward-pointing gradient can be calculated as:

$$X_{\text{att}} = \text{sign}\left(X + \frac{\delta f}{\delta \overline{x}}\right)|X| \tag{21}$$

Optimization in case of inward pointing derivative
For the inward pointing derivative we use the directional information of the derivative and then use the maximum possible perturbation in that direction. If $\phi_{\delta\overline{x}}$ denotes the phase of $\frac{\delta f}{\delta \overline{x}}$, we calculate the following value:

$$\begin{aligned} &X_{\text{att}} = X + \alpha \exp(i\phi_{\delta\overline{x}}) \\ &\text{with } \alpha > 0 \text{ s.t. } |X_{\text{att}}| = |X| \end{aligned} \tag{22}$$

This formulation has a closed form solution for α with:

$$\alpha = -2|X|\cos(\phi_{\delta\overline{x}} - \phi_X) \tag{23}$$

fulfilling $\alpha > 0$ for $|\phi_{\delta\overline{x}} - \phi_X| \in [\frac{\pi}{2}, \frac{3\pi}{2}]$, which is equivalent to the gradient pointing inwards. The proof for the closed form solution relies on arithmetic in the complex domain to simplify (22) and can be found in the appendix[2].

2) Optimization for Restricted Phase Perturbations: In the case where we can only perturb the phase in a restricted way, we determine the perturbation direction (increasing or decreasing the phase) and then perturb the phase maximally in that direction. Note that according to (18) the maximum perturbation of the phase is $2\arcsin\left(\frac{\epsilon}{2|X|}\right)$ (for those entries, that are restricted). We can determine the direction as follows:

$$\text{We have to increase the phase, if } \mathcal{I}\left(\frac{\delta f}{\delta \overline{x}}\overline{X}\right) > 0 \tag{24}$$

and vice versa. Thus, we get the perturbed phase by:

$$X_{\text{att}} = |X|\exp\left(i\phi_X + / - 2\arcsin\left(\frac{\epsilon}{2|X|}\right)\right) \tag{25}$$

where we use an addition or subtraction based on (24).

B. Magnitude Attack

To be able to analyze the impact of the Phase Attack properly, we design the contrary adversarial attack as well: The Magnitude Attack, e.g. an adversarial attack, that allows perturbation of the magnitude, while the phase stays constant.

To this end, let $X \in \mathbb{C}^n$ be an input to be perturbed with label Y, f a (real- or complex-valued) model to predict Y from X and l a suitable loss function. For the Magnitude Attack, we then have to solve:

$$\begin{aligned} &Z^* = \text{argmax}_{Z \in \mathbb{C}^n}\, l(f(Z), Y) \\ &\text{s.t. } ||Z - X||_\infty < \epsilon \text{ and } \phi_Z \underset{p.w.}{=} \phi_X \end{aligned} \tag{26}$$

This is equivalent to solving the following:

$$\begin{aligned} &Z^* = \tilde{Z}^* \exp(i\phi_X) \text{ with} \\ &\tilde{Z}^* = \text{argmax}_{\tilde{Z} \in \mathbb{R}^n_{\geq 0}}\, l(f(\tilde{Z}\exp(i\phi_X)), Y) \\ &\text{s.t. } ||\ \tilde{Z} - |X|\ ||_\infty < \epsilon \end{aligned} \tag{27}$$

$\tilde{Z} \in \mathbb{R}^n_{\geq 0}$ is the magnitude of the desired output, restricting it to be greater than or equal to zero guarantees that the phase cannot flip, i.e. that the phase is not altered by π. This is visualized in Fig. 1.

To optimize the Magnitude Attack, we once again determine the direction of an optimization step and then update the magnitude in that direction with the maximum allowed

[2]For the appendix, please refer to the code (link above).

perturbation. For an optimization step of step size ϵ this leads to:

$$X_{\text{att}} = \max\left(0, X + \epsilon \operatorname{sign}\left(\mathcal{R}\left(\frac{\delta f}{\delta \overline{x}} \overline{X}\right)\right)\right) e^{i\phi_X} \tag{28}$$

Note, that the same logic applies as in (19) and (20) to determine the direction of optimization step.

VI. Experimental Results

We conduct experiments on two image classification datasets from two different domains which often use CVNNs: the S1SLC_CVDL PolSAR [48] and FastMRI Prostate [49] (preprocessed as proposed by [50]) datasets and train a real-valued and a complex-valued ResNet [51] and ConvNeXt [52] model on each dataset, where we used initialization from RVNNs for the CVNNs [53]. We used their approach EQ without additional pretraining of the RVNNs. Both models were trained with a learning rate of 10^{-4} for 50 epochs using the Adam optimizer [54] on a cross entropy loss function with batch sizes of 64 and 200 on the Prostate and PolSAR datasets, respectively. Early stopping was used, if the validation F1 score did not improve for 10 epochs, we stop the training and use the model with the best validation F1 score for the evaluation on the test set. We trained all models with and without additional adversarial training [9] which is known to improve adversarial robustness.

With these experiments, we aim to compare adversarial robustness between RVNNs and CVNNs. As highlighted above, we are specifically interested in how perturbations of the phase information in complex-valued inputs influence the model performance. Since real-valued datasets do not possess phase information, this is a phenomenon limited to the complex domain. To get insight into those two aspects — the general adversarial robustness and specifically the robustness to Phase Attacks — we conduct two separate sets of experiments on all trained models in Section VI-A and Section VI-B. All results shown are normalized to the model performance without adversarial attacks, since we are not interested in comparing model performance, but rather robustness.

A. Comparing Robustness of Real- and Complex-Valued Models

In this experiment we compare the robustness of RVNNs and CVNNs when presented with the straightforward adaptations of adversarial attacks as presented in Section IV.

On the PolSAR dataset (Fig. 3, top row) it shows that the complex-valued ResNet and ConvNeXt are both more robust to adversarial attacks. While both RVNNs and CVNNs start to decrease when exposed to an attack restricted by $\epsilon = 0.005$ both RVNNs decrease faster. The CVNNs are more robust at all noise levels.

On the Prostate dataset (Fig. 3, second row) the results for the ResNet model are similar, all methods lead to a significant reduction in model performance with a sharp drop after $\epsilon = 0.0001$ and the CVNNs show a higher robustness to the adversarial attacks than the RVNNs. The ConvNeXt model shows the opposite behavior, for this model, the RVNNs are much more robust to the adversarial attacks at almost all noise levels.

The results on the models with additional adversarial training (Fig. 3, third and fourth row) show a similar trend. It can be observed that the model performance stays more stable for attacks with a lower attack bound, but still decreases with increasing attack strength. Especially for the Prostate dataset, adversarial training has a greater effect on the CVNNs than on the RVNNs, leading to (relatively) improved robustness on both models. On the ResNet this leads to a more robust CVNN compared to the RVNN, while on the ConvNext the increased robustness is able to equalize the superior robustness of the RVNN without adversarial training.

B. Comparing Phase, Magnitude and Regular Attacks

The results on both datasets show a similar pattern (Fig. 4). The Phase Attacks, Magnitude Attacks and the unrestricted attacks are well suited to decrease the model performance, if the strength of the attack is sufficient.

The Magnitude Attacks are generally less effective than the Phase Attacks and unrestricted attacks, following the latter's overall trends. Thus, we focus on discussing the differences between the Phase and Regular Attacks.

For both models and on both datasets the single-step Phase Attacks PFGSM and PFFGSM are less effective than their counterparts $\mathbb{C}$FGSM and $\mathbb{C}$FFGSM. Interestingly however, the multi step Phase Attacks PIFGSM and PMIFGSM are more effective than $\mathbb{C}$IFGSM and $\mathbb{C}$MIFGSM. On the PolSAR dataset the multi step Phase Attacks always accomplish the goal of completely fooling the model, i.e. reaching a score of 0, on all tasks with a maximum perturbation of 0.01 or higher, while the non-restricted base attacks do never accomplish this with any of the given perturbation strengths. On the Prostate dataset the models performance decreases even more by both attacks, sometimes already reaching a score of 0 in all cases with a perturbation strength of 0.005.

As adversarial training has no meaningful impact on the comparison of the three attacks, the results are omitted for the sake of brevity. Notably, adversarial training is also effective against the Phase and Magnitude Attacks even though it was only done with the unrestricted attacks. All observations made on the results without adversarial training still largely hold.

C. Discussion

The first set of experiments show that CVNNs are generally on par or more robust to adversarial attacks than RVNNs, regardless of adversarial training. There is only one case (ConvNeXt on prostate without adversarial training) in which the RVNN is more robust. In all other scenarios, the performance of RVNNs decrease faster than the performance of the corresponding CVNNs, when applying the same adversarial attack. Notably, generally even the weakest adversarial attack ($\mathbb{C}$FGSM) has a stronger effect on the RVNNs than the

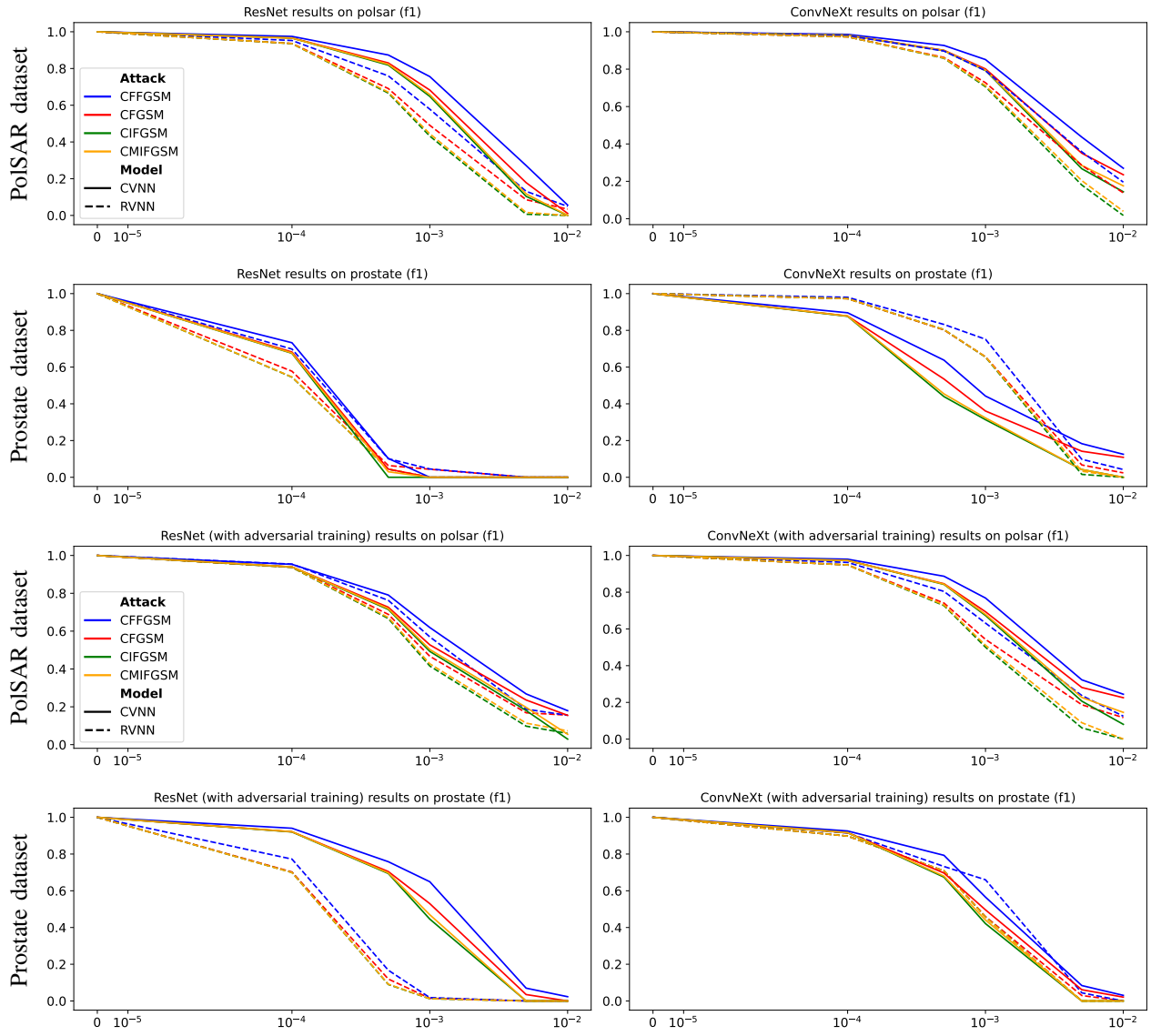

Fig. 3. Comparison of robustness against (unrestricted) adversarial attacks of RVNNs and CVNNs on the FastMRI Prostate and PolSAR datasets without (Top two rows) and with (Bottom two rows) additional adversarial training. Left: ResNet, Right ConvNeXt. All plots show (normalized) performance against ϵ-restriction. Linestyle: straight line: CVNNs, dashed line: RVNNs. Colors differentiate attack algorithms as shown in legend.

strongest attack (ℂIFGSM) on the CVNNs. In two experiments however, the RVNN is more robust than the CVNN.

The second set of experiments show that both CVNNs as well as RVNNs are highly susceptible to changes in the phase alone. Even when restricting the adversarial attacks to only alter the phase of the input, the adversarial attacks stay effective. Interestingly, in all experiments the iterative Phase Attacks showed to be more effective than the non-restricted base attacks. This is surprising, since from a theoretical standpoint the Phase Attacks are strictly weaker than the non-restricted attacks. This suggests that the restriction offered by the Phase Attacks makes it easier for the iterative optimization algorithm to find a suitable minimum. This observation is further reinforced by the results on the Magnitude Attacks against which the models are more robust than even the unrestricted attacks and therefore also the Phase Attacks.

VII. Conclusion

In this work, we have shown evidence that CVNNs can be more robust to adversarial attacks than RVNNs in some cases. To further investigate how adversarial attacks behave in the complex domain, we introduced Phase Attacks — adversarial attacks, that are limited to attack the phase of the complex-valued input but leave the magnitude unaltered. The Phase Attacks show a similar and in case of iterative optimizers even increased effectiveness when compared to regular attacks, leading to the conclusion that neural networks dealing with complex-valued data are highly susceptible to phase changes.

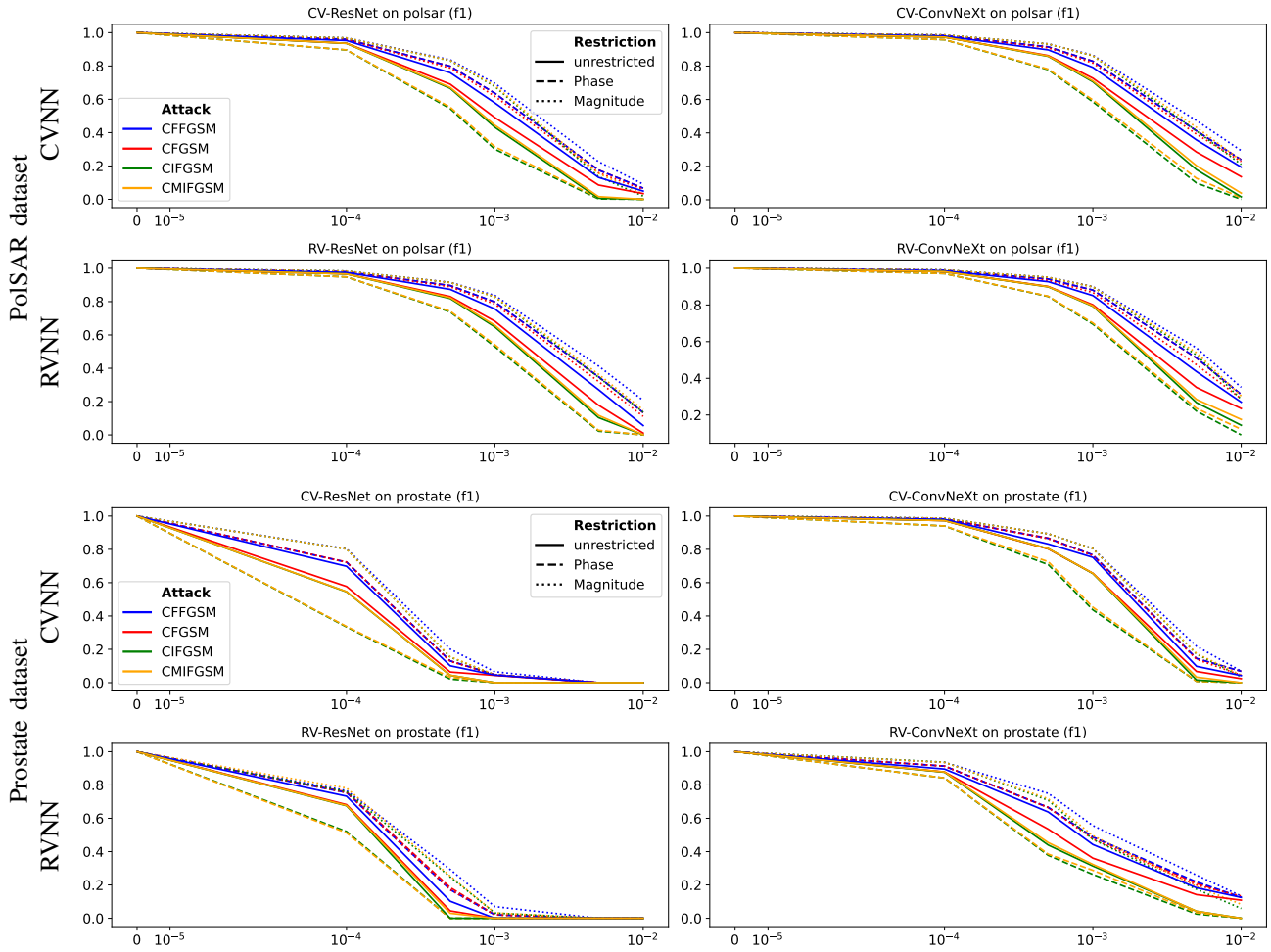

Fig. 4. Comparison of Phase Attacks and unrestricted attacks without adversarial training. Left: ResNet, Right: ConvNeXt. All plots show (normalized) performance against ϵ-restriction. Linestyle: straight line: Regular Attack, dashed line: Phase Attack, dotted Line: Magnitude Attack. Colors differentiate attack algorithms as shown in legend.

This is further reiterated by the fact, that the Magnitude Attack (which can not alter the phase at all) is less effective than the other two attack types. This should be considered when dealing with complex-valued datasets, especially in safety critical environments such as MRI in clinical practice.

As future work we would like to further investigate how CVNNs and RVNNs can be made more robust to phase changes specifically as well as test our hypothesis of susceptibility to phase perturbations on a wider scale. Additionally, the influence of changes in the phase on other definitions of robustness — such as robustness to noise or distribution changes — should be investigated.

References

[1] E. Cole, J. Cheng, J. Pauly, and S. Vasanawala, "Analysis of deep complex-valued convolutional neural networks for MRI reconstruction and phase-focused applications," *Magnetic Resonance in Medicine*, vol. 86, no. 2, pp. 1093–1109, 2021.

[2] B. Vasudeva, P. Deora, S. Bhattacharya, and P. M. Pradhan, "Compressed sensing MRI reconstruction with Co-VeGAN: Complex-valued generative adversarial network," in *WACV*, 2022.

[3] S. Moon and Y. Kim, "Mounting angle prediction for automotive radar using complex-valued convolutional neural network," *Sensors*, vol. 25, no. 2, p. 353, 2025.

[4] A. Fuchs, J. Rock, M. Tóth, P. Meissner, and F. Pernkopf, "Multiantenna radar signal interference mitigation using complex-valued convolutional neural networks," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 55, no. 3, pp. 1997–2008, 2025.

[5] K. Nguyen, C. Fookes, S. Sridharan, and A. Ross, "Complex-valued iris recognition network," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 1, pp. 182–196, 2023.

[6] Z. Zhang, Z. Lei, M. Zhou, H. Hasegawa, and S. Gao, "Complex-valued convolutional gated recurrent neural network for ultrasound beamforming," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 5668–5679, 2024.

[7] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," *arXiv preprint arXiv:1312.6199*, 2013.

[8] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," *arXiv preprint arXiv:1412.6572*, 2014.

[9] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," in *International Conference on Learning Representations*, 2018.

[10] E. Wong, L. Rice, and J. Z. Kolter, "Fast is better than free: Revisiting adversarial training," in *International Conference on Learning Representations*, 2020.

[11] Y. Dong, F. Liao, T. Pang, H. Su, J. Zhu, X. Hu, and J. Li, "Boosting

adversarial attacks with momentum," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 9185–9193.

[12] J. Lin, C. Song, K. He, L. Wang, and J. E. Hopcroft, "Nesterov accelerated gradient and scale invariance for adversarial attacks," in *International Conference on Learning Representations*, 2020.

[13] X. Wang and K. He, "Enhancing the transferability of adversarial attacks through variance tuning," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 1924–1933.

[14] C. Xie, Z. Zhang, Y. Zhou, S. Bai, J. Wang, Z. Ren, and A. L. Yuille, "Improving transferability of adversarial examples with input diversity," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 2730–2739.

[15] H. Liang, E. He, Y. Zhao, Z. Jia, and H. Li, "Adversarial attack and defense: A survey," *Electronics*, vol. 11, no. 8, p. 1283, 2022.

[16] F. Tramèr, A. Kurakin, N. Papernot, I. Goodfellow, D. Boneh, and P. McDaniel, "Ensemble adversarial training: Attacks and defenses," *arXiv preprint arXiv:1705.07204*, 2017.

[17] N. Papernot, P. McDaniel, X. Wu, S. Jha, and A. Swami, "Distillation as a defense to adversarial perturbations against deep neural networks," in *2016 IEEE symposium on security and privacy (SP)*. IEEE, 2016, pp. 582–597.

[18] J. Cohen, E. Rosenfeld, and Z. Kolter, "Certified adversarial robustness via randomized smoothing," in *international conference on machine learning*. PMLR, 2019, pp. 1310–1320.

[19] A. Athalye, N. Carlini, and D. Wagner, "Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples," in *International conference on machine learning*. PMLR, 2018, pp. 274–283.

[20] A. Shafahi, M. Najibi, M. A. Ghiasi, Z. Xu, J. Dickerson, C. Studer, L. S. Davis, G. Taylor, and T. Goldstein, "Adversarial training for free!" *Advances in neural information processing systems*, vol. 32, 2019.

[21] D. Stutz, M. Hein, and B. Schiele, "Disentangling adversarial robustness and generalization," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 6976–6987.

[22] D. Yin, R. Gontijo Lopes, J. Shlens, E. D. Cubuk, and J. Gilmer, "A fourier perspective on model robustness in computer vision," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[23] Y. L. I. N. N. Aizenberg and D. Pospelov, "About one generalization of the threshhold function," *Doklady Akademii Nauk SSSR*, vol. 196, no. 6, pp. 1287–1290, 1971.

[24] A. Hirose, *Complex-Valued Neural Networks: Theories and Applications*. World Scientific, 2003.

[25] T. Nitta, "An extension of the back-propagation algorithm to complex numbers," *Neural Networks*, vol. 10, no. 8, pp. 1391–1415, 1997.

[26] W. Yang, K. Chan, P. Chang *et al.*, "Complex-valued neural-network for direction-of-arrival estimation," *Electronics Letters*, vol. 30, no. 7, pp. 574–575, 1994.

[27] C. Trabelsi, O. Bilaniuk, Y. Zhang, D. Serdyuk, S. Subramanian, J. F. Santos, S. Mehri, N. Rostamzadeh, Y. Bengio, and C. J. Pal, "Deep complex networks," in *International Conference on Learning Representations*, 2018.

[28] M. Yang, M. Q. Ma, D. Li, Y.-H. H. Tsai, and R. Salakhutdinov, "Complex transformer: A framework for modeling complex-valued sequence," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 4232–4236.

[29] F. Eilers and X. Jiang, "Building blocks for a complex-valued transformer architecture," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2023.

[30] C. Lee, H. Hasegawa, and S. Gao, "Complex-valued neural networks: A comprehensive survey," *IEEE/CAA Journal of Automatica Sinica*, 2022.

[31] Z.-H. Tan, Y. Xie, Y. Jiang, and Z.-H. Zhou, "Real-valued backpropagation is unsuitable for complex-valued neural networks," *Advances in Neural Information Processing Systems*, vol. 35, pp. 34 052–34 063, 2022.

[32] H. Chen, F. He, S. Lei, and D. Tao, "Spectral complexity-scaled generalisation bound of complex-valued neural networks," *Artificial Intelligence*, vol. 322, p. 103951, 2023.

[33] H. Zhang, M. Gu, X. Jiang, J. Thompson, H. Cai, S. Paesani, R. Santagati, A. Laing, Y. Zhang, M. Yung *et al.*, "An optical neural chip for implementing complex-valued neural network," *Nature Communications*, vol. 12, no. 1, p. 457, 2021.

[34] Y. Bai, Y. Xu, S. Chen, X. Zhu, S. Wang, S. Huang, Y. Song, Y. Zheng, Z. Liu, S. Tan *et al.*, "Tops-speed complex-valued convolutional accelerator for feature extraction and inference," *Nature Communications*, vol. 16, no. 1, p. 292, 2025.

[35] C. Zhong, X. Sang, B. Yan, H. Li, D. Chen, X. Qin, S. Chen, and X. Ye, "Real-time high-quality computer-generated hologram using complex-valued convolutional neural network," *IEEE Transactions on Visualization and Computer Graphics*, 2023.

[36] X. Liu, J. Yu, T. Kurihara, C. Wu, Z. Niu, and S. Zhan, "Pixelwise complex-valued neural network based on 1d FFT of hyperspectral data to improve green pepper segmentation in agriculture," *Applied Sciences*, vol. 13, no. 4, p. 2697, 2023.

[37] P. Xing, J. Porée, B. Rauby, A. Malescot, E. Martineau, V. Perrot, R. L. Rungta, and J. Provost, "Phase aberration correction for in vivo ultrasound localization microscopy using a spatiotemporal complex-valued neural network," *IEEE Transactions on Medical Imaging*, 2023.

[38] A. Yakupoğlu and Ö. C. Bilgin, "Comparison of complex-valued and real-valued neural networks for protein sequence classification," *Neural Computing and Applications*, pp. 1–14, 2024.

[39] P. Virtue, X. Y. Stella, and M. Lustig, "Better than real: Complex-valued neural nets for mri fingerprinting," in *IEEE International Conference on Image Processing (ICIP)*. IEEE, 2017, pp. 3953–3957.

[40] Q. Dou, Z. Wang, X. Feng, A. E. Campbell-Washburn, J. P. Mugler III, and C. H. Meyer, "Mri denoising with a non-blind deep complex-valued convolutional neural network," *NMR in Biomedicine*, vol. 38, no. 1, p. e5291, 2025.

[41] D. A. Ron, "On the noise robustness of analog complex-valued neural networks," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 2023, pp. 37–50.

[42] D. Zhou, N. Wang, H. Yang, X. Gao, and T. Liu, "Phase-aware adversarial defense for improving adversarial robustness," in *International Conference on Machine Learning*. PMLR, 2023, pp. 42 724–42 741.

[43] A. Neacşu, R. Ciubotaru, J.-C. Pesquet, and C. Burileanu, "Design of robust complex-valued feed-forward neural networks," in *2022 30th European Signal Processing Conference (EUSIPCO)*. IEEE, 2022, pp. 1596–1600.

[44] E. C. Yeats, Y. Chen, and H. Li, "Improving gradient regularization using complex-valued neural networks," in *International Conference on Machine Learning*. PMLR, 2021, pp. 11 953–11 963.

[45] A. Kurakin, I. J. Goodfellow, and S. Bengio, "Adversarial examples in the physical world," in *Artificial intelligence safety and security*. Chapman and Hall/CRC, 2018, pp. 99–112.

[46] N. Akhtar and A. Mian, "Threat of adversarial attacks on deep learning in computer vision: A survey," *Ieee Access*, vol. 6, pp. 14 410–14 430, 2018.

[47] W. Wirtinger, "Zur formalen theorie der funktionen von mehr komplexen veränderlichen," *Mathematische Annalen*, vol. 97, no. 1, pp. 357–375, 1927.

[48] R. M. Asiyabi, M. Datcu, A. Anghel, and H. Nies, "Complex-valued end-to-end deep network with coherency preservation for complex-valued SAR data reconstruction and classification," *IEEE Transactions on Geoscience and Remote Sensing*, pp. 1–17, 2023.

[49] R. Tibrewala, T. Dutt, A. Tong, L. Ginocchio, R. Lattanzi, M. B. Keerthivasan, S. H. Baete, S. Chopra, Y. W. Lui, D. K. Sodickson *et al.*, "Fastmri prostate: A public, biparametric mri dataset to advance machine learning for prostate cancer imaging," *Scientific Data*, vol. 11, no. 1, p. 404, 2024.

[50] M. Rempe, F. Hörst, C. Seibold, B. Hadaschik, M. Schlimbach, J. Egger, K. Kröninger, F. Breuer, M. Blaimer, and J. Kleesiek, "Tumor likelihood estimation on mri prostate data by utilizing k-space information," *arXiv preprint arXiv:2407.06165*, 2024.

[51] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.

[52] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie, "A convnet for the 2020s," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 11 976–11 986.

[53] F. Eilers and X. Jiang, "Initializing complex-valued neural networks from their pretrained real-valued counterparts," in *International Joint Conference on Neural Networks (IJCNN)*, 2025.

[54] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *3rd International Conference on Learning Representations ICLR*, 2015.