# Shift-Equivariant Complex-Valued Convolutional Neural Networks

**Quentin Gabot**[1,3], **Teck-Yian Lim**[4], **Jérémy Fix**[2], **Joana Frontera-Pons**[3], **Chengfang Ren**[1] and **Jean-Philippe Ovarlez**[1,3]

[1] SONDRA, CentraleSupélec, Université Paris-Saclay, Gif-sur-Yvette, France

[2] LORIA, CNRS, CentraleSupélec, Université Paris-Saclay, France

[3] DEMR, ONERA, Université Paris-Saclay, Palaiseau, France

[4] DSO National Laboratories, Singapore

## Abstract

*Convolutional neural networks have shown remarkable performance in recent years on various computer vision problems. However, the traditional convolutional neural network architecture lacks a critical property: shift equivariance and invariance, broken by downsampling and upsampling operations. Although data augmentation techniques can help the model learn the latter property empirically, a consistent and systematic way to achieve this goal is by designing downsampling and upsampling layers that theoretically guarantee these properties by construction. Adaptive Polyphase Sampling (APS) introduced the cornerstone for shift invariance, later extended to shift equivariance with Learnable Polyphase up/downsampling (LPS) applied to real-valued neural networks. In this paper, we extend the work on LPS to complex-valued neural networks both from a theoretical perspective and with a novel building block of a projection layer from $\mathbb{C}$ to $\mathbb{R}$ before the Gumbel Softmax. We finally evaluate this extension on several computer vision problems, specifically for either the invariance property in classification tasks or the equivariance property in both reconstruction and semantic segmentation problems, using polarimetric Synthetic Aperture Radar images.*

## 1. Introduction

Since their introduction by LeCun et al. [30], Convolutional Neural Networks (CNNs) have remained the standard neural network architecture for dealing with spatially ordered data, such as images. Intuitively, many computer vision tasks rely on patterns that can appear anywhere in the image. Convolutional layers in CNNs introduce this specific bias into neural networks, as it is the 'sliding' of a kernel over the entire image. Convolutional layers are also a special case of dense linear layers, $y = Wx + b$, whereby $W$ is a Toeplitz matrix instead of any matrix.

However, modern CNN architectures include downsampling and upsampling layers that break equivariance. It has been empirically shown that resultant networks are highly sensitive to translations in the input [55]. To maintain the desirable shift equivariant property, recent works have explored provably equivariant subsampling layers [9, 44]. This search for invariance and equivariance to known symmetries in input data falls into a more general research topic: geometric deep learning [5]. This branch of deep learning explores methods for accounting for the fact that the use of geometric priors can influence the design of neural networks.

While well-developed for real-valued data, this topic is far less studied in the field of complex-valued neural networks, where both the input data and the neural network parameters are composed of complex numbers. [23, 48]. Imaging modalities commonly encountered in remote sensing (e.g., Synthetic Aperture Radar (SAR) [32]) and medical imaging (e.g., Magnetic Resonance Imaging (MRI) [13]) often consist of naturally complex-valued images. The typical approach in dealing with such data is often only utilizing the magnitude of the complex numbers, even though rich information can be embedded in the phase [4, 15, 17]. Complex-valued deep learning is an active research topic, as it has the potential to discover and exploit patterns that are either overlooked or yet to be discovered. Several foundational works had already been published [18, 51, 53].

**The key contributions of this work are:**

- The extension of provable shift-equivariance to the complex domain,
- A learnable complex-to-real projection,
- The empirical results demonstrating the benefits of a fully complex neural network for naturally complex data on three computer vision tasks (classification, reconstruction, segmentation).

In Section 2, we present the existing works related to key concepts like shift-invariant/equivariant CNNs and

CVNNs. In Section 3, we extend the theory of shift-invariant/equivariant convolutional neural networks to the complex domain by providing a theoretical explanation of their design. Section 4 describes an extensive set of experiments to demonstrate the impact of the shift-invariance/equivariance property on CVNNs. Finally, Section 5 discusses some perspectives in remote sensing [41], medical imaging [14], and earth monitoring [43].

**Notations**: Italic type indicates a scalar quantity $x$, lower case boldface indicates a vector quantity $\mathbf{v}$, and upper case boldface indicates a matrix or tensor $\mathbf{A}$. $\Re(z)$ and $\Im(z)$ respectively denote the real and imaginary parts of complex number $z \in \mathbb{C}$. $\lfloor x \rfloor$ is the mathematical floor function. The modulo operation is defined as $\mathrm{mod}$. For any vector $\mathbf{v}$, $\mathbf{v}^T$ denotes the transpose operator and $\mathbf{v}[.]$ is the indexing operator on $\mathbf{v}$. $\|\mathbf{v}\|_2$ stands for the Euclidean norm. The operator $\odot$ is the Hadamard element-wise product between vectors. Finally, the composition operator $\circ$ is defined by $(f \circ g)(x) = f(g(x))$.

## 2. Related work

**Shift-invariant/equivariant convolutional neural networks.** As discussed in Section 1, while convolution operations (using a stride of 1) are indeed shift-equivariant, modern CNNs are not. As explained by Azulay et al. [2], the traditional downsampling scheme, namely pooling layers, involving a stride strictly larger than 1, is responsible for breaking this property. We present such a case in Appendix A.

Some methods have tried to mitigate the negative effect of the subsampling scheme, for instance, by using anti-aliasing, like in the Low-Pass Filtering (LPF) method proposed by Zhang et al. [55, 56]. However, as demonstrated by Chaman et al. [9], LPF reduces the effect of downsampling layers but does not solve the shift-equivariance problem.

The first consistent shift-equivariant method was introduced by Chaman et al. with the Adaptive Polyphase Sampling (APS) method [9]. APS relies on a shift-invariant sub-sampling scheme by selecting the highest $\ell_p$-norm among the polyphase components, as formally defined in Section 3.2.

Lastly, Lim et al. [44] further elaborate on this concept and propose a generalization of APS with the Learnable Polyphase Sampling (LPS) method. Indeed, while APS relies on the $\ell_p$ norm to select the polyphase components, LPS introduces a pair of trainable <shift-invariant/equivariant><down/up-sampling>layers. This property enables an elegant method for selecting a downsampling scheme trained in conjunction with the rest of the network. Shift-invariance is desirable for classification problems, while shift-equivariance is necessary for tasks such as segmentation and reconstruction. In the latter, a translation of the input should result in the same output translation.

Shift-invariant/equivariant methods have also been extended to vision transformers (ViT) [45]. Finally, shift-equivariance is one special case of a broader research topic, group-equivariance, discussed in [12, 50].

**Complex-valued neural networks.** Complex-valued neural networks (CVNNs) are deep neural networks whose operations and inputs are defined in the complex domain [3, 23, 48]. Despite the novelty of complex-valued neural network theory, several critical questions regarding CVNNs are still open, one being their suitable representation.

Based on the isomorphism between $\mathbb{C}$ and $\mathbb{R}^2$, CVNNs can be either represented as [31]:

- A dual-RVNN is a neural network with representations in $\mathbb{R}$, where real and imaginary parts are stacked at the input. Operations are then performed in $\mathbb{R}$,
- A split-CVNN processes representations in $\mathbb{R}^2$ with operations in $\mathbb{R}$. Compared to a dual-RVNN, we keep the relationship between real and imaginary components, although the weights are also real-valued.
- A full-CVNN contains both representations and operations in $\mathbb{C}$.

However, these neural networks are not equivalent: dual-RVNNs lack any representation of the complex domain. Meanwhile, Hirose et al. [24] argue that fully-CVNNs are better representation choices for CVNNs. Finally, Wu et al. [52] recently proved that complex-valued neurons can learn more than real-valued neurons, reigniting discussions surrounding this question.

The core components of neural networks have been largely extended to the complex domain. For instance, Li et al. [34] proposed using Wirtinger calculus to define complex-valued backpropagation properly, and more recently, Trabelsi et al. [48] extended the batch normalization layer to the complex domain.

In this paper, we extend the LPS approach to CVNNs, specifically fully-CVNNs, and compare it to RVNNs, namely dual-RVNNs.

## 3. Complex-valued shift-invariant/equivariant layers

As discussed in Section 2, shift-equivariance is a highly desired property for complex-valued neural networks. While shift-equivariant methods for neural networks have been proposed and tested for the real domain, an extension of these methods to the complex domain has yet to be realized. Among the various algorithms, such as APS [9] and LPS [44], LPS is the current state-of-the-art shift-equivariant method for RVNNs. As such, our contributions focus on its extension to the complex domain. We will also introduce a learnable projection function to address some difficulties encountered during the extension of the LPS method to the complex domain.

### 3.1. Preliminaries

In this section, we formally introduce crucial concepts related to shift-invariance/equivariance and CVNNs. Note that every notion discussed in this section is defined by default in the complex domain. For the sake of simplicity, we consider, in the following, 1D signals and downsampling factor $p = 2$, although the method is generalizable for higher dimensions and downsampling factors.

**Shift-equivariance.** Shift-equivariance is a property attributed to functions whose outputs are shifted accordingly to the shift in their inputs. Formally, a function $f : \mathbb{C}^N \mapsto \mathbb{C}^M$ is $\mathbf{T}_N, \{\mathbf{T}_M, \mathbf{I}\}$-equivariant (i.e. shift-equivariant) iif:

$$\exists \mathbf{T} \in \{\mathbf{T}_M, \mathbf{I}\},\ \forall \mathbf{z} \in \mathbb{C}^N,\ f\left(\mathbf{T}_N \mathbf{z}\right) = \mathbf{T} f\left(\mathbf{z}\right), \quad (1)$$

where $\forall n \in \mathbb{Z}$, $\mathbf{T}_N \mathbf{v}[n] \triangleq \mathbf{v}\left[(n+1) \bmod N\right]$ is a circular shift.

**Shift-invariance.** The shift-invariance property refers to the case where an output is never affected by any shift in the input. Formally, a function $f : \mathbb{C}^N \mapsto \mathbb{C}^M$ is $\mathbf{T}_N, \{\mathbf{I}\}$-invariant (i.e. shift-invariant) iif:

$$f(\mathbf{T}_N \mathbf{z}) = f(\mathbf{z}), \forall \mathbf{z} \in \mathbb{C}^N. \quad (2)$$

We obtain shift-invariance from shift-equivariant functions by applying a permutation-invariant linear form (an operator mapping from $\mathbb{C}^N$ to $\mathbb{C}$ invariant under any permutation of the components of its input). One such example is the sum operation, as we observe that:

$$\sum_m f\left(\mathbf{T}_N \mathbf{z}\right)[m] = \sum_m \left(\mathbf{T}_M f(\mathbf{z})\right)[m], \quad (3)$$

is shift-invariant if $f$ is shift-equivariant. From a more general perspective, based on the above claim, any global pooling operator, i.e., an operator with a global receptive field (max, mean, sum, or product), is a shift-invariant transform.

**Complex-valued downsampling & pooling layer.** Unlike the previously introduced global pooling, pooling layers with a local receptive field are not shift-invariant/equivariant. Indeed, commonly used pooling layers are fundamentally a filter (max, mean) with a stride of 1 followed by a subsampling (using a stride strictly larger than 1), which is not a shift-invariant/equivariant operator. Indeed, a downsampling-by-$p$ factor operator $\mathbf{D}_p : \mathbb{C}^N \mapsto \mathbb{C}^{\lfloor N/p \rfloor}$ is defined as:

$$\mathbf{D}_p(\mathbf{z})[n] = \mathbf{z}[p\, n],\ \forall n \in [\![1, \lfloor N/p \rfloor]\!]. \quad (4)$$

Such an operator constantly samples the same indices of its input, regardless of whether a shift occurred.
Note that some pooling layers in the complex domain differ from their real-valued counterparts. The max pooling layer cannot be trivially extended to the complex domain as $\mathbb{C}$ is only partially ordered. As such, the returned element is a complex-valued scalar such that the selection rule is ordered, the complex-valued max pooling layer $\mathbf{M} : \mathbb{C}^N \mapsto \mathbb{C}$ being defined as:

$$\mathbf{M}(\mathbf{z}) = \mathbf{z}[k] \text{ where } k = \arg\max_{i \in [\![1,n]\!]} |\mathbf{z}[i]|, \quad (5)$$

**Complex-valued activations.** Several activation functions have been proposed in the literature: Kuroe et al. [29] proposed to distinguish between them by categorizing activation functions into the following two classes of complex functions $f(\cdot)$:

$$\begin{aligned} &\text{Type A} : f(z) = f_{\Re}(\Re(z)) + j\, f_{\Im}(\Im(z)), \\ &\text{Type B} : f(z) = \psi(r)\, e^{j\phi(\theta)}, \end{aligned} \quad (6)$$

where $f_{\Re} : \mathbb{R} \mapsto \mathbb{R}$ and $f_{\Im} : \mathbb{R} \mapsto \mathbb{R}$ are nonlinear real-valued functions. $\psi : \mathbb{R}^+ \mapsto \mathbb{R}^+$ and $\phi : \mathbb{R} \mapsto \mathbb{R}$ are also nonlinear real functions. We select the $\mathrm{modReLU}(.)$ function [1], such as given $b \in \mathbb{R}$ a bias parameter of the nonlinearity, $\mathrm{modReLU}(.)$ is defined as :

$$\begin{aligned} \mathrm{modReLU}(\mathbf{z}) &= \mathrm{ReLU}\left(|\mathbf{z}| + b\right)\, e^{j\theta} \\ &= \begin{cases} (|\mathbf{z}| + b)\, \dfrac{\mathbf{z}}{|\mathbf{z}|} & \text{if } |\mathbf{z}| + b > 0, \\ 0 & \text{otherwise}. \end{cases} \end{aligned} \quad (7)$$

**Complex differentiability.** The partially ordered nature of $\mathbb{C}$ is incompatible with minimization and optimization. Thus, loss functions $L$, estimating the correctness of the model prediction, must return a real-valued scalar ($L : \mathbb{C}^n \mapsto \mathbb{R}$). One common way to define such functions is to consider only the magnitude of $\mathbf{z}$ or compute the loss function on the real and imaginary parts separately.
Additionally, on the topic of complex differentiability, since the loss $L$ is lower bounded, it is not holomorphic by Liouville's theorem. Therefore, the Wirtinger derivative is used to compute the gradient of loss $L$ with respect to the real and imaginary parts of the neural network weights in the complex backpropagation algorithm [47].

**Complex-valued neural networks.** In addition to pooling layers, loss functions, and activation functions, further building blocks are required to achieve a complex-valued neural network. Linear and convolutional layers extend naturally to the complex domain if their weights are complex-valued. As explained by Trabelsi et al. [48], complex-valued batch normalization and initialization layers require specific implementations, and we followed their proposal in our experiments.

### 3.2. Extension of shift-equivariant methods to the complex domain

This section presents a general methodology to define shift-invariant/equivariant methods to "make convolutional networks shift-equivariant" (regardless of their real or complex-valued nature). LPF aims to replace the traditional $\mathrm{MaxPool}$, strided convolution, and $\mathrm{AvgPool}$ operations with variants that utilize low-pass filters, thereby reducing the impact of these operations on shift-invariance. However, while LPF has tried to resolve this problem, it is not a shift-invariant method but rather a good approximation of this property. To address this limit, shift-invariant APS and LPS methods have been introduced.

We now present the baseline methodology of shift-invariant methods. Let us begin by defining the polyphase component operator Poly acting on $\mathbb{C}^N$ as:

$$\mathrm{Poly}_k\left(\mathbf{z}\right)[n] = \mathbf{z}[2n+k],\ \forall n \in [\![1, \lfloor N/2 \rfloor]\!] \,,$$

where $k \in \{0,1\}$ and $\mathrm{Poly}_k(\mathbf{z}) \in \mathbb{C}^{\lfloor N/2 \rfloor}$ denotes the $k$-th *polyphase component*. We also remind

$$\mathrm{Poly}_k\left(\mathbf{T}_N\,\mathbf{z}\right) = \begin{cases} \mathrm{Poly}_1(\mathbf{z}) & \text{if } k=0\,, \\ \mathbf{T}_M \mathrm{Poly}_0(\mathbf{z}) & \text{if } k=1\,, \end{cases} \tag{8}$$

with $M = \lfloor N/2 \rfloor$. Moreover, let us denote $p_{\mathbf{z}}(K|\mathbf{z})$ as the conditional probability for selecting *polyphase components*, with $K$ denoting the Bernoulli random variable (a one-dimensional case of the categorical distribution) of polyphase indices and PD as a complex-valued Polyphase Downsampling layer, $\forall \mathbf{z} \in \mathbb{C}^N$,

$$\mathrm{PD}(\mathbf{z}) = \mathrm{Poly}_{k^*}\left(\mathbf{z}\right)\,,$$

where $k^* = \arg\max_{k\in\{0,1\}} p_{\mathbf{z}}\left(K=k|\mathbf{z}\right)$.

We can now propose **Claim 1**, **Claim 2**, and **Claim 3** which are extensions of existing real-valued claims [44].

**Claim 1:** *If $p_{\mathbf{z}}$ is shift-permutation-equivariant, then* PD *is a shift-equivariant complex-valued downsampling layer.*

We say that $p_{\mathbf{z}}$ is shift-permutation-equivariant iif:

$$p_{\mathbf{z}}\left(K=\pi(k)|\mathbf{T}_N\mathbf{z}\right) = p_{\mathbf{z}}(K=k|\mathbf{z})\,, \tag{9}$$

where $\pi(.)$ denotes the permutation on the polyphase indices $\pi(k) = k+1 \mod p$ with $p$ the number of polyphase components (downsampling factor). When there is only 2 polyphase components, $p=2$, then $\pi(1)=0$ and $\pi(0)=1$. In other words, shifting $\mathbf{z}$ leads to selecting the other polyphase component. The proof of this claim is provided in Appendix B.

For instance, consider a typical downsampling scheme, such as the strided convolution. A convolution with stride $s$ is equivalent to a convolution with stride $1$ followed by a downsampling-by-$s$ operator (see Eq. (4)). The downsampling operator (4) does not follow equality (9) as soon as $s \geq 2$, since it is guaranteed that $K=s$, regardless of whether the data $\mathbf{z}$ may or may not have been circular-shifted. It follows that the usual strided convolution is not a shift-equivariant downsampling layer.

One way to ensure that $p_{\mathbf{z}}$ is shift-equivariant is to define a shift-invariant $f : \mathbb{C}^{\lfloor N/2 \rfloor} \mapsto \mathbb{R}$ function, and an adaptive sampling layer $p_{\mathbf{z}}$ as the conditional probability:

$$p_{\mathbf{z}}(K=k|\mathbf{z}) = \frac{\exp\left(f(\mathrm{Poly}_k(\mathbf{z}))\right)}{\sum\limits_{j\in\{0,1\}} \exp\left(f(\mathrm{Poly}_j(\mathbf{z}))\right)}\,. \tag{10}$$

**Claim 2:** *If $f$ is shift-invariant, then $p_{\mathbf{z}}$ is shift-permutation equivariant.*

The proof of this claim is provided in Appendix B. Once $f$ is defined as a shift-invariant function, the actual choice of the function remains open. Many options have been proposed in the literature: the choice of function dramatically influences the neural network's performance. For instance, APS proposed to define $f(\cdot) = ||\cdot||$, implying that $k^* = \arg\max_{k\in\{0,1\}} ||\mathrm{Poly}_k(\mathbf{z})||$ since the $\mathrm{Softmax}$ operator is a smooth approximation of the $\arg\max$ operator on its input components. The polyphase component selection is based on the maximal norm value. Hence, as stated in Eq. (2), $f$ is shift-invariant (due to the global respective field of the max operator). As such, **Claim 1** and **Claim 2** are respected, proving that the downsampling layer proposed in APS is indeed shift-equivariant.

While APS hardcoded the selection rule of the polyphase components, LPS introduced a learnable rule $f_{\boldsymbol{\theta}} : \mathbb{R}^{C\times H\times W} \mapsto \mathbb{R}$ as a small convolutional neural network parametrized by $\boldsymbol{\theta}$ that learn a downsampling scheme alongside the model training. This network does not use any downsampling scheme: $f_{\boldsymbol{\theta}}$ is composed of convolutional layers with a stride $1$ and a global pooling at the output, as seen in Figure 1. Hence, as stated in Eq. (2), $f_{\boldsymbol{\theta}}$ is shift-invariant. As such, **Claim 1** and **Claim 2** are respected, proving that the downsampling layer proposed in LPS is indeed shift-equivariant. We note that LPS generalizes the APS method (as demonstrated [44, Claim 2]).

Finally, to ensure that the shift-equivariance property is correctly validated during the upsampling (for segmentation and reconstruction tasks), we denote the partial inverse polyphase component (IPoly) operator acting on $\mathbf{C}^N$ as

$\forall n \in [\![1, N]\!], \forall j \in \{0,1\}$:

$$\text{IPoly}_k(\mathbf{z})[2n+j] = \begin{cases} \mathbf{z}[n], \text{ if } j = k \\ 0, \text{ otherwise}, \end{cases} \quad (11)$$

where $k \in \{0,1\}$ and $\text{IPoly}(\mathbf{z})_k \in \mathbb{C}^{2N}$ denotes the $k$-th *upsampled component*. Note that the IPoly operator satisfies the property: $\text{Poly}_k(\text{IPoly}_k(\mathbf{z})) = \mathbf{z}$. Given a downsampled feature map $\mathbf{y} = \text{PD}(\mathbf{z}) \triangleq \text{Poly}_{k^*}(\mathbf{z})$, the complex-valued polyphase upsampling layer is defined as $\text{PU}(\mathbf{y}) = \text{IPoly}_{k^*}(\mathbf{y})$ where $k^* = \arg\max_{k\in\{0,1\}} p_{\mathbf{z}}(K = k|\mathbf{z})$. Thus, we propose the following claim:

**Claim 3:** *If $p_{\mathbf{z}}$ is shift-permutation equivariant, then* $\text{PU} \circ \text{PD}$ *is shift-equivariant.*

A mandatory condition for **Claim 3** to be valid is that the overall architecture of the model is symmetric, i.e., the number of subsampling layers in the encoder must be equal to the number of upsampling layers in the decoder. The proof of this assertion is the extension of the real-valued proof provided in [45, Claim 4]. The proof of Claim 3 is provided in Appendix B.

While PU provides a shift-equivariant upsampling scheme, it introduces zeros in the output, which results in high-frequency components. This is known as aliasing in a multirate system [49]. Similar to [44], we apply a low-pass filter scaled by the upsampling factor after each PU to resolve this.

In the following section, we propose to design a function $f_{\boldsymbol{\theta}} : \mathbb{C}^{C\times H\times W} \mapsto \mathbb{R}$ that generalizes the function $f_\theta$ proposed in APS and LPS.

## 3.3. Learnable Projection Function

To effectively learn a downsampling scheme, we parametrize $f_\theta$ as a small complex-valued neural network using Gumbel Softmax [25, 40]. The Gumbel Softmax is a differentiable method that samples from a categorical distribution using a temperature parameter, allowing it to asymptotically converge to the $\arg\max$ operation. Mathematically, we define $[x_i, \cdots, x_n]^T \in \mathbb{R}^n$ as the logits for selecting the polyphase components, $[g_i, \cdots, g_n]$ as independent samples from standard Gumbel distribution. After integration, we obtain:

$$\mathbb{P}\left(j = \arg\max_i (g_i + x_i)\right) = \frac{\exp(x_j)}{\sum_{i=1}^{n} \exp(x_i)}. \quad (12)$$

A detailed introduction of the Gumbel Softmax method is provided in Appendix C. Its extension to the complex domain requires specific attention. Logits being real-valued is a mandatory requirement for using the Gumbel Softmax, as it transforms a real-valued logit and a random Gumbel noise into a categorical distribution. To tackle this issue, we present a straightforward adaptive method to map logits from $\mathbb{C}^N$ to $\mathbb{R}^N$ in Section 3.3.

Due to the partial order of $\mathbb{C}$, extending existing operators is not straightforward. For instance, the Gumbel Softmax method requires a real-valued logit to be passed as an argument. Note that the Gumbel Softmax is not the only component that needs projecting complex-valued logits to $\mathbb{R}$; this is also the case for the cross-entropy loss function, as discussed in Section 3.1.

As projecting from $\mathbb{C}^N$ to $\mathbb{R}^N$ can be done in various ways, we propose to distinguish between two types of projection: **implicit** and **explicit projection**.

### 3.3.1. Implicit Projection

For any $\mathbf{z} \in \mathbb{C}^N$, implicit projection is realized by applying a function on $\Im(\mathbf{z})$ and $\Re(\mathbf{z})$ separately and composing the results. When dealing with implicit projection, $\text{Softmax}$ is a natural choice by computing the **Mean Softmax** $\in [0,1]^N$ and the **Product Softmax** $\in [0,1]^N$ of $\text{Softmax}(\Im(\mathbf{z}))$ and $\text{Softmax}(\Re(\mathbf{z}))$ as:

$$\text{MSoftmax}(\mathbf{z}) = \frac{1}{2}\left[\text{Softmax}(\Re(\mathbf{z})) + \text{Softmax}(\Im(\mathbf{z}))\right],$$
$$\text{PSoftmax}(\mathbf{z}) = \text{Softmax}(\Re(\mathbf{z})) \odot \text{Softmax}(\Im(\mathbf{z}))).$$

We naturally extend these two functions to obtain the equivalent for the Gumbel Softmax, i.e., $\text{MGumbelSoftmax}$ and $\text{PGumbelSoftmax}$.

### 3.3.2. Explicit Projection

Explicit projection consists of applying an additional function defined as $f : \mathbb{C}^N \mapsto \mathbb{R}^N$ on top of a neural network layer. The advantage of this approach over implicit projection is that it allows for more flexibility, due to its trainable nature. Computing the mean/product of probabilities, as done in the implicit projection approach, lacks a rigorous mathematical foundation, whereas defining a function that maps $\mathbb{C}^N$ to $\mathbb{R}^N$ is more principled.

**Norm:** For any $\mathbf{z} \in \mathbb{C}^N$, a well-known explicit projection function is the norm function, defined as $|\mathbf{z}|$. [1] Nevertheless, while this function is commonly used to project a complex number into the real domain, the phase information is discarded. This payoff might be undesirable in many cases. To address this issue, we propose a learnable projection function that generalizes the norm function while leveraging the adaptability of the connectionist paradigm.

**Polynomial:** We introduce a learnable projection polynomial layer to generalize the norm function by adapting the

[1] Using the Norm function as a projection layer for the LPS method differs from the APS method. Indeed, APS selects components using the argmax on the norm of its input.

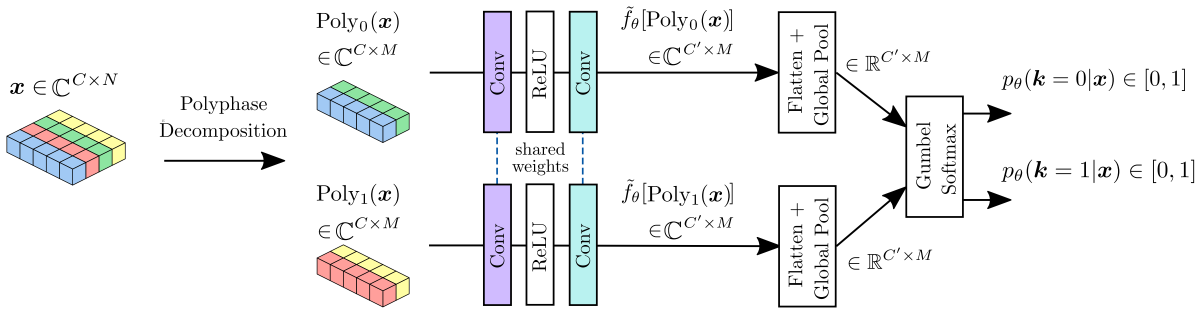


Figure 1. Proposed complex-valued extension to the shift-equivariant model first introduced in [44]. During training, we retain the weights sharing, pooling operations, and stochasticity as in [44]. We must project tensors to $\mathbb{R}$ before computing the Gumbel Softmax (or apply the Gumbel Softmax independently on the real and imaginary parts) to handle complex-valued weights and inputs.

projection function during the learning process. The proposed layer processes complex-valued input data $\mathbf{z} \in \mathbb{C}^N$ by decomposing each component $\mathbf{z}_l$ of the vector $\mathbf{z} \in \mathbb{C}^N$, for $l \in [\![1, N]\!]$, into a $M$-polynomial function acting on its real and imaginary components, $\mathbf{a}_l = \Re(\mathbf{z})_l$ and $\mathbf{b}_l = \Im(\mathbf{z})_l$, respectively. It computes all terms $\mathbf{a}_l^i\,\mathbf{b}_l^j$ where $i, j \geq 0$, $i + j = m$, for each degree $m$ ranging from 1 to $M$, the specified order of the polynomial expansion. These terms are used as inputs to a linear transformation, resulting in the output, $\forall l \in [\![1, N]\!]$:

$$\text{PolyDec}_l(\mathbf{z}) = \beta + \sum_{m=1}^{M} \sum_{i=0}^{m} \theta_{m,i}\, \mathbf{a}_l^i\, \mathbf{b}_l^{m-i}, \qquad (13)$$

where $\{\theta_{i,j} \in \mathbb{R}\}_{(i,j)\in[1,M]^2}$ are the learned weights and $\beta \in \mathbb{R}$ is the bias term of the linear layer which is for the constant term ($i = j = 0$). This formulation effectively models $f_{\boldsymbol{\theta}}$ as a polynomial function PolyDec of the real and imaginary parts of $\mathbf{z}$, up to degree $M$, capturing complex interactions between them.

**MLP:** While the amplitude is a hard-wired function and the polynomial projection function is constrained to follow an explicit formulation, we also introduce a more general class of learnable projection functions as an MLP. We define the multilayer perceptron (MLP) as a sequence of fully connected layers that process the real and imaginary parts of the complex input $\mathbf{z}$, such as:

$$\text{MLP}(\mathbf{z}) = f^{(L)} \circ f^{(L-1)} \circ \ldots \circ f^{(1)} \left( \begin{bmatrix} \Re(\mathbf{z}) \\ \Im(\mathbf{z}) \end{bmatrix} \right), \qquad (14)$$

where $L$ is the number of layers in the network, each layer $f^{(l)}$ for $l \in [\![1, L]\!]$ is defined as a real-valued non-linear transformation.
Based on the universal approximation theorem, this layer can generalize all possible projection functions without being constrained to a specific polynomial form. Nevertheless, note that its size constrains the generalization capabilities of the MLP.

## 4. Experiments

In this section, we evaluate the performance of complex-valued shift-invariant/equivariant neural networks in the complex domain. More precisely, we demonstrate how complex-valued shift-invariant/equivariant neural networks outperform traditional complex-valued neural networks across various computer vision tasks and possess an inherent superiority in their design, particularly in terms of the impact of the shift on the input. We also test our approach on dual-RVNNs to evaluate the effect of the complex nature of CVNNs compared to their real-valued counterparts. We conduct experiments on image classification, semantic segmentation, and image reconstruction. To align our experiments with the scope of this article, we carefully selected complex-valued imaging datasets upon which our models were tested. Indeed, by only experimenting to the data's physical background. While we reproduce the existing setup for circular shift [9], we propose the first setup to evaluate complex-valued shift-invariant/equivariant neural networks in this section. More precisely, we used the Cr. S. metric from [9, 44, 55] to evaluate shift-equivariance/invariance by applying a random circular shift between 1 and 9 on the input of a model, and its output. Shift-equivariant/invariant models should achieve a 100% perfect score, unlike non-equivariant/invariant models.
However, we note that outperforming existing state-of-the-art models is outside the scope of this work, as we primarily focus on assessing the exact impact of each component of our method. To this end, we tested our methods on complex-valued adaptations of standard SOTA CNNs architectures (ResNet [21], UNet [46], AutoEncoder[22]). We detail each task's architectures and training setup in Appendix F and provide a complete guide to reproduce our experiments at the following repository[2].

[2] https://github.com/QuentinGABOT/Equivariant_CVNN

### 4.1. Datasets

While experiments presented in this section are based on computer vision tasks, the dataset of interest concerns polarimetric SAR (PolSAR) imagery [32]. This imaging technique is based on the coherent combination of radar echoes, yielding an image of the ground scene. The SAR system is usually mounted on a satellite or an aircraft, providing complex-valued information about the backscatter characteristics of the ground. Each pixel of a PolSAR image is associated with a complex scattering matrix $\mathbf{S}$, called the Sinclair matrix $\mathbf{S} = \begin{pmatrix} S_{HH} & S_{VH} \\ S_{HV} & S_{VV} \end{pmatrix}$ where the indices $H$ and $V$ refer to the horizontal and vertical polarization states, but any orthogonal polarimetry basis can be used. PolSAR datasets are presented in more detail in Appendix E, while polarimetric decompositions are detailed in Appendix D.
Note that even if the following results have been obtained with PolSAR images, other types of complex-valued imaging could have been used, notably magnetic resonance imaging (MRI) [35]. We specifically recognize that CVNNs have been utilized for specific MRI applications, such as fat-water separation and flow quantification [13].

| Model | Type | OA (%) | F1 (%) | Cr. S. (%) |
|---|---|---|---|---|
| ResNet LPS MLP | $\mathbb{C}$ | **84.97** | **80.62** | **100.0** |
| ResNet LPS | $\mathbb{R}$ | 82.92 | 74.49 | **100.0** |
| ResNet LPS PolyDec | $\mathbb{C}$ | 81.63 | 73.5 | **100.0** |
| ResNet LPS MSoftmax | $\mathbb{C}$ | 79.5 | 72.23 | **100.0** |
| ResNet LPS Norm | $\mathbb{C}$ | 76.38 | 68.25 | **100.0** |
| ResNet APS | $\mathbb{C}$ | 82.01 | 75.37 | **100.0** |
| ResNet APS | $\mathbb{R}$ | 81.16 | 73.08 | **100.0** |
| ResNet LPF | $\mathbb{C}$ | 81.56 | 73.68 | 93.5 |
| ResNet LPF | $\mathbb{R}$ | 79.66 | 70.8 | 95.44 |

Table 1. Classification and circular shifts metrics on the S1SLC_CVDL dataset. Shift-equivariant models are separated from non-shift-equivariant models, namely CVNN/RVNN Low-Pass Filter (LPF), by the additional row. CVNN Learnable Polyphase Sampling (LPS) with MLP projection layer outperforms all variants, even Adaptive Polyphase Sampling (APS).

### 4.2. Classification

To test our approach on the classification task, we propose an experiment setup based on the S1SLC_CVDL dataset, as described in Appendix F.1. On average, we report for an A100 GPU, the following training times per epoch: CVNN LPS PolyDec: $32m30s$, CVNN LPF: $22m15s$, RVNN LPF: $06m06s$. We also report the number of trainable parameters and the inference time for the shift-equivariant CVNN variants in Appendix F.1. Results from Table 1 validate our initial hypothesis that complex-valued shift-invariant CVNNs outperform their non-shift-invariant counterparts.
Our proposed approach, a CVNN LPS with an MLP projection function, outperforms every other model. Unlike non-shift-invariant models, shift-invariant CVNNs and RVNNs score a perfect circular shift consistency. Furthermore, the three best-performing models (when looking at the Overall Accuracy (OA) and the F1 score) are shift-invariant, highlighting this property's benefits. Nevertheless, we note that not *all* shift-invariant models outperform non-shift-invariant models based on the OA and F1 score: this interesting result likely indicates that the design choice behind the downsampling scheme and the projection function matters. More specifically, the APS approach consistently appears to be a worse choice than the LPS strategy, indicating that this non-learnable downsampling scheme is outperformed by its learnable variant. This observation can also be made when comparing our different projection functions: the Norm function is also generalized and outperformed by the PolyDec and the MLP. Finally, we observe that RVNNs *almost always* underperformed against their CVNN counterparts. We believe this result comes from the arduous classification of classes from the S1SLC_CVDL dataset. Naturally, exploiting the phase information might confer CVNNs a unique advantage over RVNNs. We provide additional results and visualizations in Appendix G.1.

| Model | Type | OA (%) | F1 (%) | Cr. S. (%) |
|---|---|---|---|---|
| UNet LPS PolyDec | $\mathbb{C}$ | **97.21** | **92.7** | **100.0** |
| UNet LPS | $\mathbb{R}$ | **97.21** | 91.86 | **100.0** |
| UNet LPS Norm | $\mathbb{C}$ | 96.04 | 89.47 | **100.0** |
| UNet LPS MLP | $\mathbb{C}$ | 95.24 | 89.32 | **100.0** |
| UNet LPS MSoftmax | $\mathbb{C}$ | 94.59 | 87.05 | **100.0** |
| UNet APS | $\mathbb{R}$ | 96.73 | 90.15 | **100.0** |
| UNet APS | $\mathbb{C}$ | 93.78 | 83.85 | **100.0** |
| UNet LPF | $\mathbb{R}$ | 97.13 | 91.62 | 97.01 |
| UNet LPF | $\mathbb{C}$ | 93.12 | 84.58 | 97.04 |

Table 2. Semantic segmentation and circular shifts metrics on the PolSF dataset. Shift-equivariant models are separated from non-shift-equivariant models, namely CVNN/RVNN Low-Pass Filter (LPF), by the additional row. CVNN Learnable Polyphase Sampling (LPS) with PolyDec projection layer outperforms all variants, even Adaptive Polyphase Sampling (APS).

### 4.3. Semantic Segmentation

We propose an experiment setup based on the PolSF dataset, as described in Appendix F.2, to test our approach to the semantic segmentation task. On average, we report the following training times per epoch for the A100 GPU: CVNN LPS PolyDec: 38 seconds, CVNN LPF: 34 seconds, RVNN LPF: 14 seconds. We also report the number of trainable parameters and the inference time for the shift-equivariant CVNN variants in Appendix F.2. Table 2 shows that the results validate our initial hypothesis that complex-valued shift-equivariant CVNNs outperform their non-shift-equivariant counterparts.

Results show that the proposed approach outperforms all other models, specifically a CVNN LPS with a PolyDec projection function. Unlike non-shift-equivariant models, shift-equivariant CVNNs and RVNNs score a perfect circular shift consistency. Furthermore, the two best-performing models are shift-equivariant with respect to performance metrics, like OA and F1 score. Nevertheless, we note that not *all* shift-equivariant models outperform non-shift-equivariant models based on the OA and F1 score; thus, we reiterate the same conclusions as in Section 4.2. Finally, we observe that RVNNs *almost always* outperform their CVNN counterparts. Contrary to the observations made in Section 4.2, we believe that this might be because the semantic segmentation of the PolSF dataset is a rather simple task (notably due to the relatively small size of the dataset): the phase information might not be a sufficient argument to use CVNNs over RVNNs in this case. Nonetheless, the CVNN LPS PolyDec appears to be a striking exception to this observation: this leads us to hypothesize that CVNNs may have a more challenging time learning some representations than RVNNs, but they ultimately possess a more substantial representation capability if correctly designed. We provide additional results and visualizations in Appendix G.2.

### 4.4. Reconstruction

Finally, to test our approach on the reconstruction task, we propose an experiment setup based on the San Francisco Polarimetric SAR ALOS-2 dataset, as described in Appendix F.3. The polarimetric properties of the data allow us to evaluate the reconstruction using additional metrics by computing coherent and non-coherent polarimetric decompositions on the reconstructed image. We leverage these decompositions to assess the model's performances: namely, the Pauli [10, 33], Krogager [26–28], Cameron [6–8], and $H-\alpha$ [11, 16] decompositions, as presented in Appendix D. Then, as is usually done by the signal processing community, we compute classification metrics from the Cameron decomposition and the $H-\alpha$ decomposition. Please note that the following metrics are obtained from an unsupervised process meant to be calculated on a fully trained model on the reconstruction task [17]. More details regarding the experiments can be found in Appendix G.3. On average, we report the following training times per epoch for the A100 GPU: CVNN LPS PolyDec: 22 seconds, CVNN LPF: 13 seconds, RVNN LPF: 2 seconds. We also report the number of trainable parameters and the inference time for the shift-equivariant CVNN variants in Appendix F.3. Finally, we did not apply the low-pass filter for the reconstruction task, as it does not show any improvements. Results from Table 3 validate our initial hypothesis that complex-valued shift-equivariant CVNNs outperform their non-shift-equivariant counterparts. We can also notice the pattern observed in Section 4.2, i.e., that CVNNs outperform their RVNN counterparts. Even though we believe these results are logical, as polarimetric decompositions (such as $H-\alpha$) are highly sensitive to amplitude and phase values, these results highlight that the CVNNs are particularly well-suited for the reconstruction task, which is even more accentuated when they are shift-equivariant. Furthermore, we notice a clear distinction in performance between the shift-equivariant models (the first four, as opposed to the last three), further validating our hypothesis that the choice of both downsampling and projection layers matters.

| Model | Type | MSE (↓) | H-$\alpha$ OA (%) | H-$\alpha$ F1 (%) | Cam. OA (%) | Cam. F1 (%) | Cr. S (↓) |
|---|---|---|---|---|---|---|---|
| AE LPS PolyDec | $\mathbb{C}$ | $\mathbf{1.3 \times 10^{-4}}$ | **96.66** | **95.56** | **94.03** | **91.40** | **0.0** |
| AE APS | $\mathbb{C}$ | $1.5 \times 10^{-4}$ | 95.50 | 93.88 | 93.05 | 90.23 | **0.0** |
| AE LPS MLP | $\mathbb{C}$ | $3.6 \times 10^{-4}$ | 94.54 | 92.67 | 90.41 | 86.74 | **0.0** |
| AE LPS Norm | $\mathbb{C}$ | $4.2 \times 10^{-4}$ | 94.33 | 92.37 | 89.62 | 86.04 | **0.0** |
| AE APS | $\mathbb{R}$ | $1.1 \times 10^{-2}$ | 60.53 | 51.13 | 58.40 | 45.56 | **0.0** |
| AE LPS MSoftmax | $\mathbb{C}$ | $1.2 \times 10^{-2}$ | 58.05 | 49.17 | 59.17 | 50.87 | **0.0** |
| AE LPS | $\mathbb{R}$ | $2.2 \times 10^{-2}$ | 53.38 | 42.20 | 46.03 | 34.40 | **0.0** |
| AE LPF | $\mathbb{C}$ | $6.3 \times 10^{-2}$ | 46.97 | 32.69 | 23.74 | 17.01 | 72.21 |
| AE LPF | $\mathbb{R}$ | $8.5 \times 10^{-2}$ | 34.15 | 19.01 | 16.26 | 11.33 | 43.12 |

Table 3. Reconstruction, $H-\alpha$ classification metrics, and Cameron classification metrics on the San Francisco Polarimetric SAR ALOS-2 dataset. Shift-equivariant models are separated from non-shift-equivariant models, namely CVNN/RVNN Low-Pass Filter (LPF), by the additional row. CVNN Learnable Polyphase Sampling (LPS) with PolyDec projection layer outperforms all variants, even Adaptive Polyphase Sampling (APS). Note that the Cr. S metric differs from the supervised learning task, as we cannot measure the agreement between the predictions. As such, we compute the $\ell_2$ norm: in this case, shift-equivariant models achieve a perfect score, while non-shift-equivariant models obtain non-zero values.

## 5. Conclusion

This work extended a provably shift-equivariant convolutional neural network framework to the complex domain. In doing so, we enabled the incorporation of this strong prior to convolutional CNNs. In our experiments, we observed that LPS demonstrated better results than APS, demonstrating that a learnable downsampling scheme has an advantage over a handcrafted one. Following this observation, we introduced learnable functions PolyDec and MLP to perform the needed projection from $\mathbb{C}$ to $\mathbb{R}$. This end-to-end trainable approach allows the model to adapt effectively to the data it is trained on. In our experiments, we observe that the PolyDec projection function, and even more so, the MLP projection function, increase computational requirements and latency. This is even more aggravating when dealing with CVNNs, as they are slower to train than their RVNN counterparts. Thus, we believe the PolyDec projection function strikes a sweet intermediate between a non-learnable projection function, like Norm, and an unrestricted one, like MLP.

## References


[1] M. Arjovsky, A. Shah, and Y. Bengio. Unitary evolution recurrent neural networks. In *International conference on machine learning*, pages 1120–1128. PMLR, 2016. 3

[2] A. Azulay and Y. Weiss. Why do deep convolutional networks generalize so poorly to small image transformations? *Journal of Machine Learning Research*, 20(184):1–25, 2019. 2

[3] J. Barrachina, C. Ren, C. Morisseau, G. Vieillard, and J.-P. Ovarlez. Complex-valued vs. real-valued neural networks for classification perspectives: An example on non-circular data. In *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2990–2994, 2021. 2

[4] J. Barrachina, C. Ren, C. Morisseau, G. Vieillard, and J.-P. Ovarlez. Comparison between equivalent architectures of complex-valued and real-valued neural networks-application on polarimetric SAR image segmentation. *Journal of Signal Processing Systems*, 95(1):57–66, 2023. 1

[5] M. Bronstein, J. Bruna, Y. LeCun, A. Szlam, and P. Vandergheynst. Geometric deep learning: going beyond Euclidean data. *IEEE Signal Processing Magazine*, 34(4):18–42, 2017. 1

[6] W. Cameron and L. Leung. Feature motivated polarization scattering matrix decomposition. In *IEEE International Conference on Radar*, pages 549–557. IEEE, 1990. 8, 13

[7] W. Cameron and H. Rais. Conservative polarimetric scatterers and their role in incorrect extensions of the cameron decomposition. *IEEE transactions on Geoscience and Remote Sensing*, 44(12):3506–3516, 2006.

[8] W. Cameron, N. Youssef, and L. Leung. Simulated polarimetric signatures of primitive geometrical shapes. *IEEE Transactions on Geoscience and Remote Sensing*, 34(3): 793–803, 2002. 8, 13

[9] A. Chaman and I. Dokmanic. Truly shift-invariant convolutional neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3773–3783, 2021. 1, 2, 6

[10] S. Cloude and E. Pottier. A review of target decomposition theorems in radar polarimetry. *IEEE transactions on geoscience and remote sensing*, 34(2):498–518, 1996. 8, 12

[11] S. Cloude and E. Pottier. An entropy based classification scheme for land applications of polarimetric sar. *IEEE transactions on geoscience and remote sensing*, 35(1):68–78, 2002. 8

[12] T. Cohen and M. Welling. Group equivariant convolutional networks. In *International conference on machine learning*, pages 2990–2999. PMLR, 2016. 2

[13] 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*, 86(2):1093–1109, 2021. 1, 7

[14] M. Dedmari, S. Conjeti, S. Estrada, P. Ehses, T. Stöcker, and M. Reuter. Complex fully convolutional neural networks for mr image reconstruction. In *International Workshop on Machine Learning for Medical Image Reconstruction*, pages 30–38. Springer, 2018. 2

[15] V. Dhédin, J. Levi, J. Fix, C. Ren, and I. Hinostroza. Complex-valued wasserstein gan for sar image generation. In *IGARSS 2024 - 2024 IEEE International Geoscience and Remote Sensing Symposium*, pages 6991–6996, 2024. 1

[16] P. Formont, F. Pascal, G. Vasile, J.-P. Ovarlez, and L. Ferro-Famil. Statistical classification for heterogeneous polarimetric SAR images. *IEEE Journal of selected topics in Signal Processing*, 5(3):567–576, 2010. 8, 13

[17] Q. Gabot, J. Fix, J. Frontera-Pons, C. Ren, and J.-P. Ovarlez. Preserving polarimetric properties in polsar image reconstruction through complex-valued auto-encoders. In *2024 International Radar Conference (RADAR)*, pages 1–6, 2024. 1, 8

[18] P. Geuchen and F. Voigtlaender. Optimal approximation using complex-valued neural networks. *Advances in Neural Information Processing Systems*, 36:1681–1737, 2023. 1

[19] D. Geudtner, R. Torres, P. Snoeij, M. Davidson, and B. Rommen. Sentinel-1 system capabilities and applications. In *2014 IEEE geoscience and remote sensing symposium*, pages 1457–1460. IEEE, 2014. 14

[20] E. Gumbel. Statistical theory of extreme valuse and some practical applications. *Nat. Bur. Standards Appl. Math. Ser. 33*, 1954. 12

[21] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 6

[22] G. Hinton and R. Zemel. Autoencoders, minimum description length and helmholtz free energy. *Advances in neural information processing systems*, 6, 1993. 6

[23] A. Hirose. Complex-valued neural networks: The merits and their origins. In *2009 International Joint Conference on Neural Networks*, pages 1237–1244, Atlanta, Ga, USA, 2009. IEEE. 1, 2

[24] A. Hirose and S. Yoshida. Generalization characteristics of complex-valued feedforward neural networks in relation to signal coherence. *IEEE Transactions on Neural Networks and learning systems*, 23(4):541–551, 2012. 2

[25] E. Jang, S. Gu, and B. Poole. Categorical reparameterization with Gumbel-Softmax. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. 5

[26] E. Krogager. Utilization and interpretation of polarimetric data in high resolution radar target imaging. In *Proc. Second International Workshop on Radar Polarimetry (JIPR'1992)*, pages 547–557, Nantes, France, 1992. 8, 12

[27] E. Krogager and Z. Czyz. Properties of the sphere, diplane, helix decomposition. In *Proc. Third International Workshop on Radar Polarimetry (JIPR'1995)*, pages 106–114, Nantes, France, 1995.

[28] E. Krogager, J. Dall, and S. Madsen. The sphere, diplane, helix decomposition recent results with polarimetric SAR data. In *Proc. Third International Workshop on Radar Polarimetry (JIPR'1995)*, pages 621–625, Nantes, France, 1995. 8, 12

[29] Y. Kuroe, M. Yoshid, and T. Mori. On activation functions for complex-valued neural networks—existence of energy

functions. In *International Conference on Artificial Neural Networks*, pages 985–992. Springer, 2003. 3

[30] Y. Lecun and Y. Bengio. Convolutional networks for images, speech, and time-series. In *The Handbook of Brain Theory and Neural Networks*, 1995. 1

[31] C. Lee, H. Hasegawa, and S. Gao. Complex-valued neural networks: A comprehensive survey. *IEEE/CAA Journal of Automatica Sinica*, 9(8):1406–1426, 2022. 2

[32] J. Lee and E. Pottier. *Polarimetric Radar Imaging, From Basics to Applications*. CRC Press, 2009. 1, 7, 12

[33] J. Lee and E. Pottier. *Polarimetric radar imaging: from basics to applications*. CRC press, 2017. 8, 12

[34] H. Li and T. Adalı. Complex-valued adaptive signal processing using nonlinear functions. *EURASIP Journal on Advances in Signal Processing*, 2008:1–9, 2008. 2

[35] Z. Liang, P. Lauterbur, IEEE Engineering in Medicine, and Biology Society. *Principles of Magnetic Resonance Imaging: A Signal Processing Perspective*. Wiley, 2000. 7

[36] T. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017. 14

[37] X. Liu, L. Jiao, F. Liu, D. Zhang, and X. Tang. Polsf: Polsar image datasets on san francisco. In *International Conference on Intelligence Science*, pages 214–219. Springer, 2022. 14

[38] I. Loshchilov and F. Hutter. Decoupled weight decay regularization, 2019. 14, 15

[39] C. Maddison, D. Tarlow, and T. Minka. A* sampling, 2015. 12

[40] C. Maddison, A. Mnih, and Y. Teh. The concrete distribution: A continuous relaxation of discrete random variables. *arXiv preprint arXiv:1611.00712*, 2016. 5, 12

[41] A. Mian, J.-P. Ovarlez, A. Atto, and G. Ginolhac. Design of new wavelet packets adapted to high-resolution sar images with an application to target detection. *IEEE Transactions on Geoscience and Remote Sensing*, 57(6):3919–3932, 2019. 2

[42] R. Mohammadi Asiyabi, M. Datcu, A. Anghel, and H. Nies. S1SLC_CVDL: A complex-valued annotated single look complex Sentinel-1 SAR dataset for complex-valued deep networks, 2023. 14

[43] J. Patruno, N. Dore, M. Crespi, and E. Pottier. Polarimetric multifrequency and multi-incidence sar sensors analysis for archaeological purposes. *Archaeological Prospection*, 20(2): 89–96, 2013. 2

[44] R. Rojas-Gomez, T. Lim, A. Schwing, M. Do, and R. Yeh. Learnable polyphase sampling for shift invariant and equivariant convolutional networks. *Advances in Neural Information Processing Systems*, 35:35755–35768, 2022. 1, 2, 4, 5, 6, 11

[45] R. Rojas-Gomez, T. Lim, M. Do, and R. Yeh. Making vision transformers truly shift-equivariant. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5568–5577, 2024. 2, 5

[46] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*, pages 234–241. Springer, 2015. 6

[47] A. Sarroff, V. Shepardson, and M. Casey. Learning representations using complex-valued nets. *arXiv preprint arXiv:1511.06351*, 2015. 3

[48] C. Trabelsi, O. Bilaniuk, Y. Zhang, D. Serdyuk, S. Subramanian, J. Santos, S. Mehri, N. Rostamzadeh, Y. Bengio, and C. Pal. Deep complex networks. *arXiv preprint arXiv:1705.09792*, 2017. 1, 2, 3

[49] M. Vetterli, J. Kovačević, and V. Goyal. *Foundations of signal processing*. Cambridge University Press, 2014. 5

[50] M. Weiler and G. Cesa. General e (2)-equivariant steerable cnns. *Advances in neural information processing systems*, 32, 2019. 2

[51] J. Wu, S. Zhang, Y. Jiang, and Z. Zhou. Complex-valued neurons can learn more but slower than real-valued neurons via gradient descent. *Advances in Neural Information Processing Systems*, 36:23714–23747, 2023. 1

[52] J. Wu, S. Zhang, Y. Jiang, and Z. Zhou. Complex-valued neurons can learn more but slower than real-valued neurons via gradient descent. *Advances in Neural Information Processing Systems*, 36, 2024. 2

[53] S. Yadav and K. Jerripothula. Fccns: Fully complex-valued convolutional networks using complex-valued color model and loss function. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 10655–10664, 2023. 1

[54] Y. Yamaguchi, R. Sato, and H. Yamada. ALOS-2 quad. pol. images and ALOS ones. In *Proceedings of EUSAR 2016: 11th European Conference on Synthetic Aperture Radar*, pages 1–4, 2016. 13

[55] R. Zhang. Making convolutional networks shift-invariant again. In *International conference on machine learning*, pages 7324–7334. PMLR, 2019. 1, 2, 6

[56] X. Zou, F. Xiao, Z. Yu, Y. Li, and Y. Lee. Delving deeper into anti-aliasing in convnets. *International Journal of Computer Vision*, 131(1):67–81, 2023. 2