Sur-Real: Frechet Mean and Distance Transform for Complex-Valued Deep Learning

Rudrasis Chakraborty, Jiayun Wang, and Stella X. Yu

UC Berkeley / ICSI

{rudra,peterwg,stellayu}@berkeley.edu

Abstract

We develop a novel deep learning architecture for naturally complex-valued data, which is often subject to complex scaling ambiguity. We treat each sample as a field in the space of complex numbers. With the polar form of a complex-valued number, the general group that acts in this space is the product of planar rotation and non-zero scaling. This perspective allows us to develop not only a novel convolution operator using weighted Fréchet mean (wFM) on a Riemannian manifold, but also to a novel fully connected layer operator using the distance to the wFM, with natural equivariant properties to non-zero scaling and planar rotations for the former and invariance properties for the latter.

We demonstrate our method on two widely used complex-valued datasets: SAR dataset MSTAR and RadioML dataset. On MSTAR data, without any preprocessing, our network can achieve 98% *classification accuracy on this highly imbalanced dataset using only* $44,000$ *parameters, as opposed to* 94% *accuracy with more than* $500,000$ *parameters with a baseline real-valued network on the two-channel real representation of the complex-valued data. On RadioML data, we get comparable classification accuracy with the baseline with only using* 10% *of the parameters as the baseline model.*

1. Introduction

We study the task of extending deep learning to naturally complex-valued data, where useful information is intertwined in both magnitudes and phases. For example, synthetic aperture radar (SAR), magnetic resonance (MR) images and radio frequency (RF) signals are acquired in complex numbers, with the magnitude often encoding the amount of energy and the phase indicating boundaries or geometrical shapes. Even for real-valued images, there are complex representations that are known to be successful for many pattern recognition tasks; the most notable examples are the Fourier transforms and spectrum-based computer vision techniques ranging from steerable filters [10] to spectral graph embedding [15, 22].

A straightforward solution is to treat the complex-valued data as two-channel real-valued data, and apply real-valued deep learning. Such an Euclidean space embedding would not respect the intrinsic geometric property of complex-valued data. For example, in both MR and SAR images, the pixel intensity value often has an ambiguity of complex-valued scaling. Of course, one can get around such an ambiguity by training data augmentation [14, 7], but such extrinsic data manipulation is time-consuming and ineffective. Ideally, deep learning on such images should be invariant to the group of non-zero scaling and planar rotations in the complex plane.

We treat each complex-valued data sample as a field in the space of complex numbers – a special non-Euclidean space. This perspective allows us to develop novel convoluational and fully connected layer functions that provide such an invariant characterization.

The major hurdle in extending the definition of convolution in the Euclidean space to a non-Euclidean space is the lack of vector space structure. Given a point on the non-Euclidean space, a translation of the point may not remain on that space. The translation equivariance property thus does not make sense on these spaces. On the other hand, in the Euclidean space, the group of translation is the group that transitively acts. Given two points in the Euclidean space, there exists a translation to go from one point to another. However, in a non-Euclidean space, this is not true, e.g., on a sphere, one can go from one point to another by rotation, not by translation.

There is a long line of works that propose definitions of convolutions in a non-Euclidean space by treating each data sample as a function in that space [21, 5, 6, 9, 3, 13].

Our key insight is that with the polar form of a complex number, the general group that acts on this space is the product of planar rotation and non-zero scaling. Essentially, we want to define a convolution operator that is equivariant to the action of this product group. When each sample is a field on a Riemannian manifold, [4] shows that:

- The convolution operator defined by weighted Fréchet mean (wFM) [17] (wFM) is equivariant to the group that naturally acts on that manifold.
- Non-linear activation functions such as ReLU may not needed. Since wFM is non-linear and acts like a contraction mapping [16] analogous to ReLU or sigmoid function. We could still use tangent ReLU for better accuracy.

Such a neural network equipped with wFM filtering on complex-valued data has a group invariant property similar to the standard CNN on real-valued data. There are some research on developing a CNN in the complex-valued domain [2, 20], but none of these works study the properties of the convolution operator, e.g., equivariance, linearity etc. Our definition of convolution has all the desired properties and is a theoretically justified analog of the real-valued CNN.

Most excitingly, on the publicly available SAR dataset MSTAR, we demonstrate the effectiveness of our approach by significant accuracy improvement with substantially fewer parameters, over the real-valued baseline model applied to two-channel real valued representation of complex-valued data. Furthermore, we apply our proposed method to radio frequency data, RadioML and shown comparable performance with fewer parameters.

To summarize, we make the following major contributions.

1. We develop a novel CNN architecture in the complex-valued domain.
2. We prove in theory that our proposed method has equivariance and invariance properties.
3. We validate our method on MSTAR, a widely used complex-valued image classification dataset. Without any preprocessing, we achieve $\sim 98\%$ classification accuracy on this highly imbalanced dataset using only 10% of parameters of the baseline model. We hypothesize that the improvement in performance is due to the usage of intrinsic definition of convolution in the complex space as opposed to the standard CNN on 2D Euclidean embedding of complex numbers.
4. We also extend our wFM convolution on popularly used radio frequency data. We showed that we can achieve comparable performance with fewer parameters.

2. A Convolutional Neural Network on Complex-Valued Data

We start this section by first present the geometry of the manifold of complex numbers, denoted by $\mathbf{C}$. Then, we develop a convolutional neural network (CNN) framework for complex-valued data. Before developing this framework, we will talk about key properties of a CNN architecture, specifically, **(a)** the *equivariance* property of a convolution operator **(b)** the *invariance* property of a CNN . We will also point out the implications of these key properties.

Space of complex numbers: A (smooth) manifold of complex numbers, $\mathbf{C}$ consists of elements of the form $a+ib$, where $a, b \in \mathbf{R}$. This manifold is a Riemannian manifold [1] and the distance induced by the canonical Riemannian metric is given by:

$$d(a+ib, c+id) = \sqrt{(a-c)^2 + (b-d)^2} \tag{1}$$

Now, we talk about the identification of $\mathbf{C}$ with it's polar form. As expected, we can show that this identification is a bijection. In the rest of the paper, we will use this polar form to represent a complex number. Now, we formally define this polar representation as:

Definition 1. *We identify each complex number, $a+ib$, with it's polar form, i.e., $r\exp i\theta$, where r and θ are the absolute value/ magnitude (abs) and argument (arg) of $a+ib$. Hence, we can identify $\mathbf{C}$ as $\mathbf{R}^+ \times \mathit{SO}(2)$, where $\mathit{SO}(2)$ is the manifold of planar rotations. Let $F : \mathbf{C} \to \mathbf{R}^+ \times \mathit{SO}(2)$ be the mapping which is given by*

$$a+ib \mapsto \left(r, \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix}\right),$$

where, $r = \mathsf{abs}(a+ib)$ and $\theta = \mathsf{arg}(a+ib)$.

Observe that F is a bijection mapping. The induced distance from Eq. (1) is given as follows. Given, $\mathbf{z}_1, \mathbf{z}_2 \in \mathbf{C}$, let, $(r_1, R_1) = F(\mathbf{z}_1)$ and $(r_2, R_2) = F(\mathbf{z}_2)$. Then,

$$d(\mathbf{z}_1, \mathbf{z}_2) = \sqrt{\log\left(\frac{r_1}{r_2}\right)^2 + \|\mathsf{logm}\left(R_1^{-1}R_2\right)\|_F^2}, \tag{2}$$

where, logm is the matrix logarithm. Furthermore with the above identification, $\mathbf{C}$ is a Riemannian homogeneous space [11]. The next two propositions is crucial for the construction of convolution operator. Before proving the next proposition, we restate some definition borrowed from group theory literature.

Definition 2. *[8]*

1. *Given a (Riemannian) manifold $\mathcal{M}$ and a group G, we say that G acts on $\mathcal{M}$ (from left) if there exists a mapping $L : \mathcal{M} \times G \to \mathcal{M}$ given by $(X, g) \mapsto g.X$ satisfies **(a)** $L(X, e) = e.X = X$ **(b)** $(gh).X = g.(h.X)$.*

2. *An action is called a transitive action iff given $X, Y \in \mathcal{M}$, $\exists g \in G$, such that $Y = g.X$.*

Proposition 1. *Using the identification in Def.* (1)*, the group, $G := \{\mathbf{R} \setminus \{0\}\} \times \mathit{SO}(2)$ transitively acts on $\mathbf{C}$ and the action is given by $((r, R), (r_g, R_g)) \mapsto (r_g r R_g, R_g R)$.*

Proof. The proof follows from the definition of a Riemannian homogeneous space [11]. □

Now that we know the group G that acts on $\mathbf{C}$, we show that the group G is the set of isometries on $\mathbf{C}$ as stated in the following proposition.

Proposition 2. *Given $\mathbf{z}_1 = (r_1, R_1), \mathbf{z}_2 = (r_2, R_2) \in \mathbf{C}$ and $g = (r_g, R_g) \in G$, $d(g.\mathbf{z}_1, g.\mathbf{z}_2) = d(\mathbf{z}_1, \mathbf{z}_2)$.*

Proof. The proof follows from Eq. (2) as

$$\begin{aligned} d(g.\mathbf{z}_1, g.\mathbf{z}_2) = \Bigg(& \log\left(\frac{r_g r_1 r_g}{r_g r_2 r_g}\right)^2 + \\ & \|\mathsf{logm}\left(R_1^{-1} R_g^{-1} R_g R_2\right)\|_F^2 \Bigg)^{0.5} \\ & = d(\mathbf{z}_1, \mathbf{z}_2) \end{aligned}$$

□

With the setup of basic properties of complex numbers needed to define convolution operator and in turn define a CNN on $\mathbf{C}$, we first illustrate the aforementioned two properties of a CNN. We first show that our definition of convolution and CNN satisfy the above two properties and then show the similarity of the definition with the standard Euclidean convolution operator.

Equivariance property of convolution: In the Euclidean convolution definition, i.e., defined on $\mathbf{R}^n$, the convolution operator is equivariant to translation, i.e., fixing the kernel of convolution if the input signal is translated by $\mathbf{t}$, the convolution output is going to be translated by $\mathbf{t}$ as well. This is a desirable property for standard convolution as it enables to share weights across the entire domain (image for most computer vision applications). But a natural question to ask is *What so special about translation?* Observe that the group of translations is the group of isometries for $\mathbf{R}^n$ and moreover transitively acts on $\mathbf{R}^n$ (as defined in Def. (2)).

From Props. (1), (2), we know that on $\mathbf{C}$, $G = \{\mathbf{R} \setminus \{0\}\} \times \mathsf{SO}(2)$ transitively acts and is the group of isometries, hence in order the generalization of the Euclidean convolution operator on $\mathbf{C}$, we need to define an operator which is equivariant to the action of G. This motivates us to explore a definition of convolution operator suitable for $\mathbf{C}$. Recently in [4], the authors proposed a CNN framework on manifold valued data. They defined a convolution operator on a manifold $\mathcal{M}$ which is equivariant to the group, G, that acts on $\mathcal{M}$. In this setting, $\mathcal{M} = \mathbf{C}$ and $G = \mathsf{SO}(2) \times \mathbf{R} \setminus \{0\}$ and hence we will define the convolution operator on $\mathbf{C}$ as follows.

Convolution operator: We will use *weighted Fréchet mean (FM)* (wFM) [17] to define convolution layer. Given $\{\mathbf{z}_i\}_{i=1}^K \subset \mathbf{C}$ and $\{w_i\}_{i=1}^K \subset (0, 1]$ with $\sum_i w_i = 1$, (the) *weighted Fréchet mean (FM)* (wFM) is defined as:

$$\mathsf{wFM}(\{\mathbf{z}_i\}, \{w_i\}) = \arg\min_{\mathbf{m} \in \mathbf{C}} \sum_{i=1}^K w_i d^2(\mathbf{z}_i, \mathbf{m}), \quad (3)$$

Here, d is the distance defined in Eq. (2). In the above definition wFM can be regarded as the minimizer of the weighted variance. Though it is posed as a optimization problem, we will use a provably convergent "cheap" alternative as proposed in [4]. Notice that in the above defined convolution operator, $\{w_i\}$ is the filter which we will learn and $\mathsf{wFM}(\{\mathbf{z}_i\}, \{w_i\}) \in \mathbf{C}$ is the convolution output. The filter $\{w_i\}$ will be learned through stochastic gradient descent. As w_i is real valued, one can use the standard SGD, but additionally we need to ensure the convexity constraint on $\{w_i\}$.

Now, we will formally state the equivariance property of the convolution operator and then justify the choice of wFM by drawing some analogy from standard convolution operator.

Proposition 3 ([4])**.** *The convolution definition as given in Eq.* (3) *is equivariant to the action of $G = \{\mathbf{R} \setminus \{0\}\} \times \mathit{SO}(2)$.*

Proof. The proof is a consequence of Prop. (2). □

In Fig (1), we have shown the equivariance property with respect to rotation and scaling.

Why wFM?: To justify the choice of wFM as convolution operator in Eq. (3), we first remind the readers to the definition of standard convolution operator. Notice that the standard convolution operator can be written as $\sum_i w_i x_i$, where $\{w_i\}$ is the filter and $\{x_i\}$ is the signal. Now notice that with the convexity constraint on $\{w_i\}$, $\sum_i w_i x_i$ is the wFM on the Euclidean space as it is the minimizer of the weighted variance as defined in Eq. (3). Now, observe that the convexity constraint is to ensure that the resultant stays on the space $\mathbf{C}$. Thus the choice of wFM as the convolution operator though looks arbitrary at first, is an obvious choice if we look at the Euclidean convolution operator as the minimizer of weighted variance. Now that we have a definition of convolution on hand, we will define a non-linear activation function. Though in [4], the authors argued that as the convolution operator defined in Eq. (3) is non-linear and is a *contraction mapping*[4], we found out that using a ReLU like activation helps with the learning. We call this activation function **tangent ReLU** which we will define next.

tangent ReLU (tReLU): Similar to the ReLU operator on $\mathbf{R}^n$, tReLU is a function from $\mathbf{C}$ to $\mathbf{C}$ as is de-

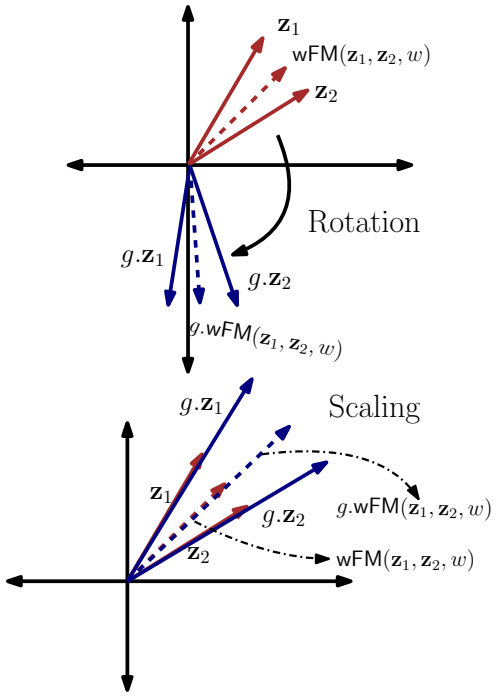

Figure 1: Equivariance with respect to (a) rotation (b) scaling in the complex plane

fined as: $(r, R) \mapsto \mathsf{Exp}\left(ReLU\left(\mathsf{log}(r), \mathsf{logm}(R)\right)\right)$, where, $\mathsf{Exp}\left(a, B\right) = \left(\mathsf{exp}(a), \mathsf{expm}(B)\right)$. expm is the matrix exponential operator.

With the convolution operator and a non-linear operator in our hand, we are now ready to define a deep convolutional network. Notice that analogous to standard CNN, we want our complex-valued CNN to be invariant (the remaining desired property of a CNN) which is our next topic of discussion.

Invariance property of CNN: In standard Euclidean CNN, the entire network is invariant to the action of the group of translations. This ensures that the output does not change if the input is translated. This invariance is achieved by the last softmax fully connected (FC) layer. It is easy to show that the standard softmax FC layer is invariant to translation. Similar to our earlier discussion, on $\mathbf{C}$ we want the CNN to be invariant to the action of G. We will now define the invariant last layer and will show that this layer is invariant to the action of G.

Invariant last layer: Let $\{\mathbf{t}_i\}_{i=1}^d \subset \mathbf{C}$ be the output of the last convolution layer, where d is the number of channels of the last convolution layer. Then, we define the last layer with inputs $\{\mathbf{t}\}_{i=1}^d$ and outputs $\{y_i\}_{i=1}^c$, where, $\{y_i\}_{i=1}^c = \mathsf{FC}\left(\{u_{i,j}\}_{i=1,j=1}^{d,l}\right)$. Here, FC is the standard fully connected + softmax layer and

$$u_{i,j} = d(\mathbf{t}_i, \mathbf{t}_u^j), \tag{4}$$

where, $\mathbf{t}_u^j = \mathsf{wFM}\left(\{\mathbf{t}_i\}, \left\{v_i^j\right\}\right)$. We will learn the filter $\left\{v_i^j\right\}$ as before. Essentially in this last layer, we will learn l number of wFMs of the input $\{\mathbf{t}_i\}_{i=1}^d$ and look at the distance from each t_i to the wFMs. This essentially gives us a cluster structure/ topology of $\{\mathbf{t}_i\}_{i=1}^d$ which is the input of the standard fully connected + softmax layer. As obvious from the construction itself, we can show that this layer is invariant to the action of $G = \{\mathbf{R} \setminus \{0\}\} \times \mathsf{SO}(2)$ as formally stated next.

Proposition 4 ([4])**.** *The above defined last layer is invariant to the action of G.*

In Fig (2), we have shown the invariance property with respect to G of our proposed ComplexNet.

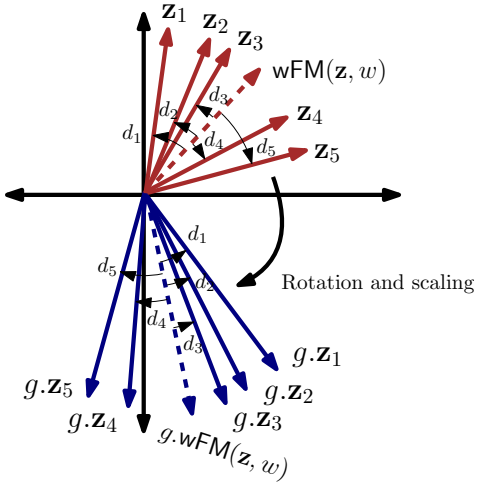

Figure 2: Invariance with respect to action of G

ComplexNet: With the above building blocks, we have a complex-valued CNN framework (dubbed as ComplexNet) which is invariant to the action of G. A schematic description of this network is given in Fig. (3). A basic building block of ComplexNet with two convolution layers is presented in Alg. (1).

3. Experimental Results

In this section, we present a proof of concept experiments on a widely used complex-valued dataset, MSTAR dataset [12] and a radio frequency data, RadioML [18, 19]. While, MSTAR is a data suitable for classification task and consists of complex-valued 2D images, RadioML is a complex-valued 1D signal appropriate for classification. For MSTAR, as a baseline, we will use the baseline architecture (with batch normalization) (as shown in Fig. (8)) with the following data embedding 1. embed a complex-valued image as a $\mathbf{R}^2$ valued image 2. take absolute value of a complex-valued

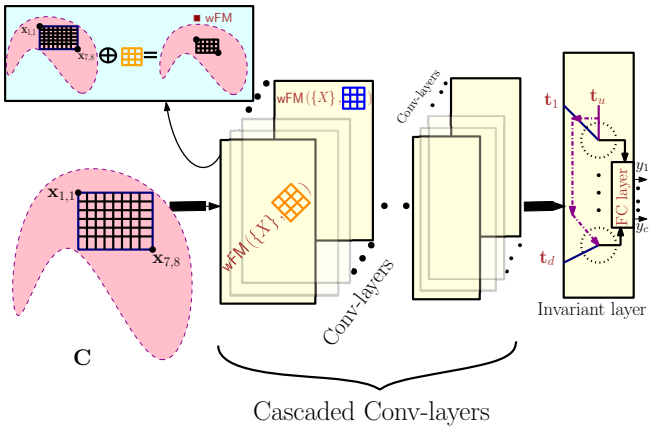

Figure 3: Schematic diagram on ComplexNet

Algorithm 1: A basic ComplexNet building block with two convolution layers

function COMPLEXNET
VARIABLES($c^1_{in}, c^1_{out}, k_1, c^2_{out}, k_2, l, c$)
$x \leftarrow Input(c^1_{in}, h, w)$
$x \leftarrow Conv(x, c^1_{out}, k_1)$
$x \leftarrow tReLU(x)$
$x \leftarrow Conv(x, c^2_{out}, k_2)$
$x \leftarrow tReLU(x)$
$x \leftarrow Inv(x, l, c)$
end function

image to make a $\mathbf{R}$ valued image. 3. $\mathbf{R}^3$ embedding with magnitude and $\mathbf{R}^2$ 4. polar (r, θ) embedding We perform a 30-70 random train test split and report the average classification accuracy over 10 runs. For RadioML, we used the baseline architecture proposed in [18, 19]. Below, we first give a brief description of the MSTAR data and then give the details of the experimental setup.

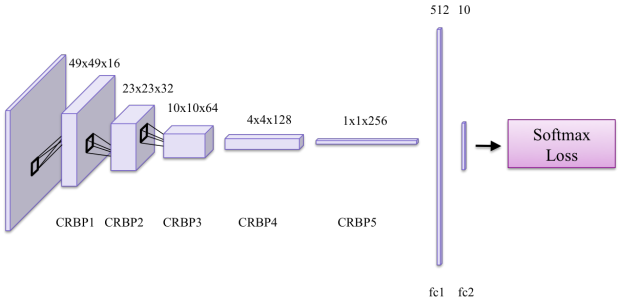

Figure 4: Baseline model MSTAR data (CRBP denoted Conv, ReLU, Batch-Normalization, Pooling)

MSTAR dataset: This dataset consists of X-band SAR image chips with 0.3m $\times$ 0.3m resolution of 10 target classes

	bmp2	btr70	t72	btr60	2s1	brmd2	d7	t62	zil131	zsu23-4
bmp2	**1257**	1	24	0	2	1	0	0	0	0
btr70	6	**418**	1	3	1	0	0	0	0	0
t72	24	1	**6624**	2	9	8	0	1	1	24
btr60	19	8	5	**407**	7	5	0	0	0	0
2s1	0	2	21	0	**1122**	12	0	0	0	7
brmd2	0	0	6	0	2	**1399**	1	0	0	7
d7	0	0	0	0	0	0	**573**	0	0	0
t62	0	0	28	0	0	0	1	**540**	3	0
zil131	0	0	7	0	0	0	3	0	**563**	0
zsu23-4	0	0	12	0	2	1	0	0	0	**1386**

Table 1: Confusion matrix for proposed ComplexNet (classification accuracy $\mathbf{98.16}\%$)

	bmp2	btr70	t72	btr60	2s1	brmd2	d7	t62	zil131	zsu23-4
bmp2	**1086**	27	11	150	0	8	0	2	1	0
btr70	1	**336**	0	91	0	0	0	1	0	0
t72	33	3	**6304**	60	13	10	0	255	3	13
btr60	0	3	0	**448**	0	0	0	0	0	0
2s1	9	19	5	53	**951**	72	0	53	1	1
brmd2	2	0	0	75	1	**1332**	0	0	5	0
d7	0	0	24	2	0	7	**507**	12	11	10
t62	0	0	44	25	1	1	0	**501**	0	0
zil131	0	0	24	7	3	3	0	3	**533**	0
zsu23-4	2	0	125	33	0	115	8	43	5	**1070**

Table 2: Confusion matrix for baseline on the embedding in $\mathbf{R}^2$ (classification accuracy 89.77%)

including BMP2 (infantry combat vehicle), BTR70 (armored personnel carrier) etc.. We cropped 100×100 region from the center of each sample image. This dataset is highly unbalanced with the number of samples per class varies in the range of 429 - 6694. Some sample images are shown in Fig. (5). Because of the highly imbalance nature of this dataset, the classification task becomes hard. We use two complex convolution layers with kernel size 5×5 and stride 5 followed by one complex convolution layer with kernel size 4×4 and stride 4, then we use an invariant last layer with a softmax layer at the end for classification. For the three complex convolution layers, the number of output channels are 50, 100 and 200 respectively. We use Adam optimizer with learning rate 0.005 and mini-batch size of 100. The confusion matrices with overall classification accuracies are given in Tables (1)-(3).

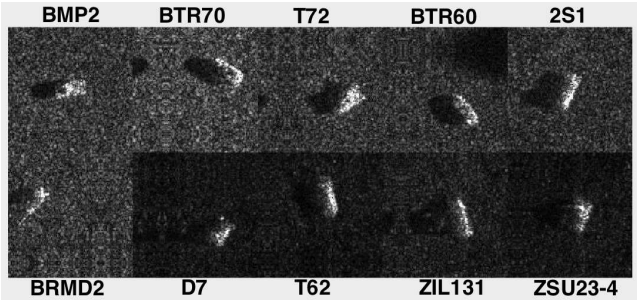

Figure 5: Sample MSTAR images

From these tables, we can see that our proposed ComplexNet performs a 3.6% *performance improvement over*

	bmp2	btr70	t72	btr60	2s1	brmd2	d7	t62	zil131	zsu23-4
bmp2	**1224**	51	7	0	3	0	0	0	0	0
btr70	0	**423**	3	0	3	0	0	0	0	0
t72	26	6	**6643**	0	3	4	0	6	1	5
btr60	4	295	21	**100**	8	2	0	21	0	0
2s1	1	40	13	0	**1094**	12	0	1	0	3
brmd2	41	8	4	0	5	**1336**	1	2	14	4
d7	0	0	1	0	0	0	**566**	0	1	5
t62	0	0	123	0	14	0	0	**432**	1	2
zil131	0	0	17	0	6	0	2	0	**544**	4
zsu23-4	0	0	9	0	0	0	3	0	0	**1389**

Table 3: Confusion matrix for baseline on the magnitude (classification accuracy 94.46%)

	bmp2	btr70	t72	btr60	2s1	brmd2	d7	t62	zil131	zsu23-4
bmp2	**1246**	1	11	6	7	13	0	0	1	0
btr70	15	**388**	0	19	4	3	0	0	0	0
t72	9	0	**6592**	0	8	4	0	16	7	58
btr60	7	1	1	**437**	1	3	1	0	0	0
2s1	1	0	4	4	**1132**	15	0	2	5	1
brmd2	3	0	0	0	0	**1407**	1	0	13	1
d7	1	0	0	0	0	0	**567**	0	1	4
t62	1	0	45	0	2	0	2	**496**	19	7
zil131	0	0	0	0	1	1	13	0	**558**	0
zsu23-4	4	0	8	0	6	1	81	0	1	**1300**

Table 4: Confusion matrix for baseline on the $\mathbf{R}^3$ embedding (classification accuracy 96.87%)

	bmp2	btr70	t72	btr60	2s1	brmd2	d7	t62	zil131	zsu23-4
bmp2	**1178**	2	23	0	56	6	1	15	3	0
btr70	22	**370**	1	3	33	0	0	0	0	0
t72	14	0	**6478**	0	36	0	0	124	21	21
btr60	43	61	0	**253**	76	8	3	6	1	0
2s1	1	1	15	0	**1125**	1	0	13	7	1
brmd2	2	2	8	0	46	**1331**	2	0	19	5
d7	0	0	0	0	0	0	**571**	0	1	1
t62	0	0	63	0	2	0	0	**492**	14	1
zil131	0	0	0	0	0	0	6	1	**565**	1
zsu23-4	0	0	94	0	2	1	26	8	20	**1250**

Table 5: Confusion matrix for baseline on the (r, θ) embedding (classification accuracy 93.51%)

	bmp2	btr70	t72	btr60	2s1	brmd2	d7	t62	zil131	zsu23-4
bmp2	**1229**	4	37	10	2	3	0	0	0	0
btr70	11	**405**	1	10	2	0	0	0	0	0
t72	43	1	**6556**	1	30	18	1	3	11	30
btr60	13	9	7	**409**	9	4	0	0	0	0
2s1	0	3	17	3	**1103**	30	0	1	3	4
brmd2	0	0	8	0	7	**1392**	1	0	2	5
d7	0	0	0	0	0	0	**571**	0	2	10
t62	0	0	39	0	4	1	0	**521**	5	2
zil131	0	1	4	0	0	0	2	0	**566**	0
zsu23-4	0	0	21	0	5	4	1	1	1	**1368**

Table 6: Confusion matrix for proposed ComplexNet using normalized complex numbers (classification accuracy $\mathbf{97.00}\%$)

the baseline. We claim that this is because of the group equivariant property of convolution and the group invariant proposed network architecture. Note that, the group that acts on complex numbers is $\mathsf{SO}(2) \times \mathbf{R} \setminus \{0\}$. This essentially means that our learned ComplexNet is invariant to scaling and planar rotations. This is unlike to any traditional CNN architecture. Furthermore this natural representation is somewhat robust to the imbalanced data which can be seen from Tables (1)-(3). We can see that for the smallest class with number of samples to be 429 (for class 'BTR70'), ComplexNet correctly classifies 406 samples while VGG16 correctly classifies only 172 sample points. This result is consistent for all the classes with small sample sizes. So, our framework essentially gives a representation which is robust to imbalanced class sizes.

For the baseline, we can see that just the magnitude gives better classification accuracy than the two channels real, $\mathbf{R}^2$ embedding. To further investigate the usefulness of magnitude, we have done an experiment with both magnitude and $\mathbf{R}^2$ in a $\mathbf{R}^3$ embedding. This $\mathbf{R}^3$ embedding achieves a classification accuracy of 96.87%, which is around 2% improvement over just using the magnitude. The confusion matrix for the $\mathbf{R}^3$ embedding is given in Table (4).

As in our ComplexNet, we used magnitude and phase, (r, R) representation, we have used the baseline model to test on this representation as well. More specifically, as a matrix R is represented by a θ, we use the polar (r, θ) representation and run our baseline model taking it as two channel reals. The confusion matrix is given in Table (5). From the table, we can see that the result is better than two channel reals, but still worse than the magnitude only representation. So, in one hand we can see that just magnitude gives very good result and can help to increase the performance of $\mathbf{R}^2$ embedding. But, with both magnitude and phase (in (r, θ) embedding), we get better accuracy than $\mathbf{R}^2$ embedding. This analysis raises the following question :*is phase information really necessary?*, which we will answer next.

What just phase can achieve: Having established the fact that the proposed ComplexNet performs well over the standard CNN architecture, now we focus on how much gain our proposed framework can achieve just from the phase information of the data. In order to see the effect, we normalize each complex number to magnitude 1 and look at the performance of ComplexNet compared to our baseline CNN using $\mathbf{R}^2$ embedding. The confusion matrix using ComplexNet is shown in Table (6). The baseline CNN using $\mathbf{R}^2$ embedding of the normalized complex numbers results 45.98% classification accuracy. A closed investigation of the confusion matrix (shown in Table (7)) reveals that all the points have been classified in the largest class which consists of 45.98% samples of the entire dataset. Recall that this result is using $30 - 70\%$ train test split. We experiment with increasing the training partition without any significant improvement. Thus, we conclude that with $\mathbf{R}^2$ embedding, using just phase information gives very poor classification accuracy. On the other hand, the performance of ComplexNet using just phase information is significant and indicate that effectiveness of ComplexNet is because of the G invariance property.

A summary classification accuracy of different variants of methods has been shown in Fig. (6).

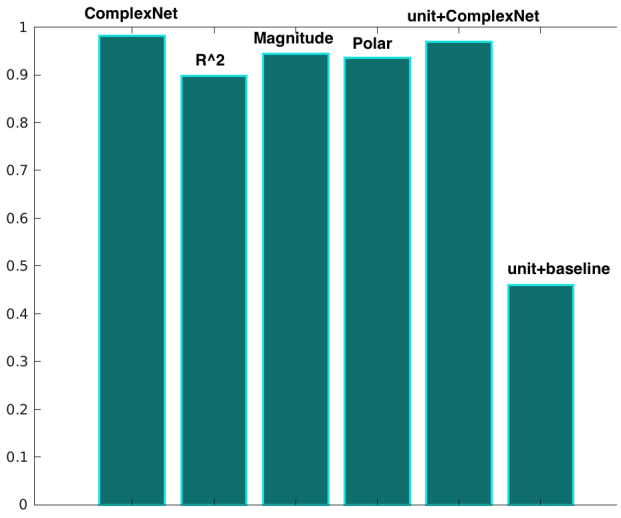

Figure 6: Barplot of classification accuracies for different variants on MSTAR

	bmp2	btr70	t72	btr60	2s1	brmd2	d7	t62	zil131	zsu23-4
bmp2	**0**	0	1285	0	0	0	0	0	0	0
btr70	0	**0**	429	0	0	0	0	0	0	0
t72	0	0	**6694**	0	0	0	0	0	0	0
btr60	0	0	451	**0**	0	0	0	0	0	0
2s1	0	0	1164	0	**0**	0	0	0	0	0
brmd2	0	0	1415	0	0	**0**	0	0	0	0
d7	0	0	573	0	0	0	**0**	0	0	0
t62	0	0	572	0	0	0	0	**0**	0	0
zil131	0	0	573	0	0	0	0	0	**0**	0
zsu23-4	0	0	1401	0	0	0	0	0	0	**0**

Table 7: Confusion matrix for baseline method using normalized complex numbers (classification accuracy $\mathbf{45.98}\%$)

Model	domain	# params.
ComplexNet	$\mathbf{C}$	**44826**
baseline	$\mathbf{R}^2$	530170
baseline	$\mathbf{R}$ (magnitude)	530026
baseline	$\mathbf{R}^3$	530314

Table 8: Comparative analysis of number of parameters

The parameter values in Table (8) indicates that we have a significant parameter reduction in our model. We hypothesized that this is because of the power of our model to capture the natural equivariance/ invariance which standard CNN fails to do in the non-Euclidean domain. This justifies the usefulness of the proposed model on complex-valued domain both in terms of parameter efficiency and classification accuracy.

In Fig. (7), we have shown the representative outputs of three convolution layers for representative samples from each class. We can see that after the first convolution layer, the output is basically similar to blurry input images as shown in Fig. (5). From the output of the second convolution layer, we can see that 10 filters output patterns are different for

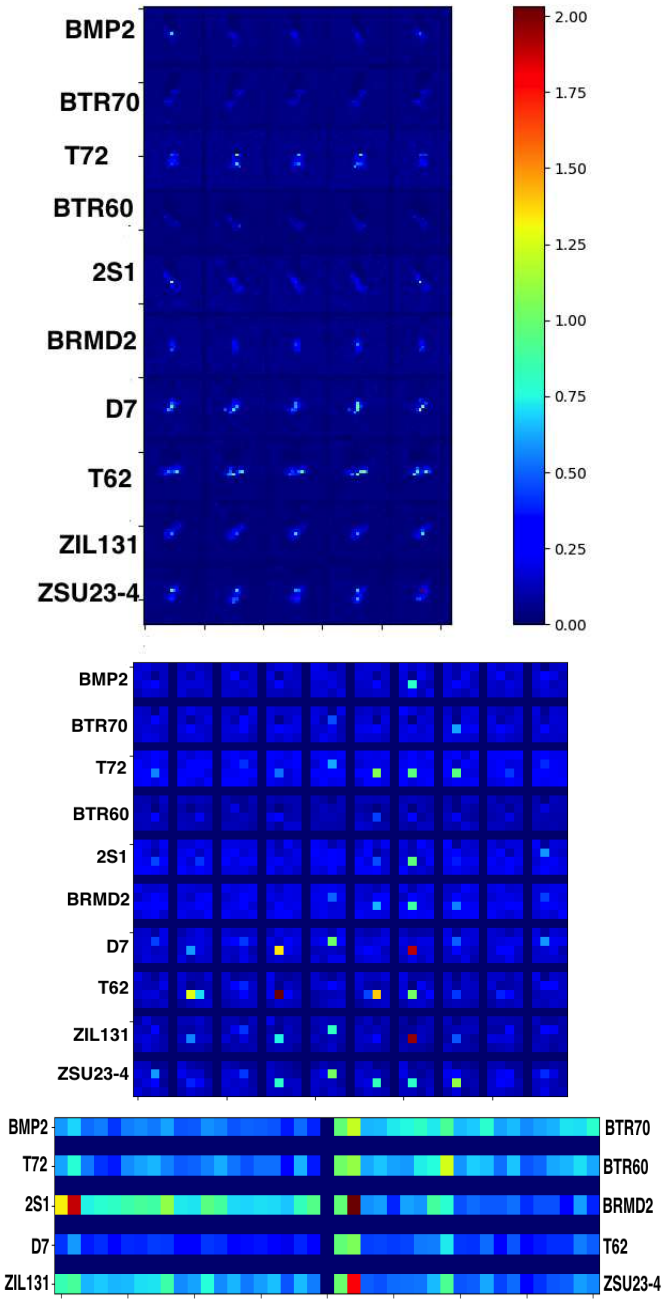

Figure 7: *(Top-Bottom)* Representative filter outputs after the first, second and last conv. layer of ComplexNet

different classes. Note that, here we have shown one representative layer output from each class, but the pattern within each class is similar. For classes 'D7', 'T62', 'ZIL131', the filter responses are higher than the other classes. Furthermore, the output of last convolution layer shows significantly different patterns between different classes.

RadioML data In this section, we show classification accuracy of ComplexNet on another complex-valued data, namely RadioML data as proposed in [18, 19]. This dataset

consists of modulations which are used widely in practice and operate on both discrete binary alphabets (digital modulations), and continuous alphabets (analog modulations). Then over each modem the known data is modulated and then exposed to the channel effects described above using GNU Radio. Finally they were segmented in millions of samples into a dataset consisting of numerous short-time windows in a fashion similar to how a continuous acoustic voice signal is typically windowed for voice recognition tasks. For a detailed data description, please see [18, 19]. This dataset consists of $220,000$ samples. We use a $50-50$ train-test split analogous to out MSTAR data experiment as mentioned in the paper.

For baseline method, we have used the architecture proposed in [18], which consists of two convolutional and two fully connected layers. The convolution layers have kernel of size 3 with number of channels 256 and 80 respectively. In between convolution layers there are ReLU and dropout layers. This network consists of 2830491 parameters.

We have used our ComplexNet with only 299117 parameters, i.e., roughly 10% of the number of parameters of the baseline model. We have achieved 70.23% and 70.68% testing accuracy for ComplexNet and baseline model respectively. This result is consistent with out previous finding, that with significantly less number of parameters, ComplexNet can achieve similar result as the baseline network. Finally, we have shown filter responses using ComplexNet in Fig. (8). In each row, we have shown sample filter responses from each class. We can see from the figure that filter outputs of the convolution layers are different over different classes.

4. Conclusions and Future directions

In this paper, we have presented a novel convolutional neural network on complex-valued data, dubbed as ComplexNet. We have shown theoretical properties of the proposed network as well as the defined convolution operator. More specifically, we have shown that our proposed convolution operator is equivariant and the proposed ComplexNet is invariant to the product group of planar rotations and non-zero scaling. Experimental results on widely used MSTAR dataset have shown that the proposed ComplexNet achieves very high classification accuracy using a very small number of parameters. As a baseline, we have compared both using the $\mathbf{R}^2$ embedding and by taking magnitude of the complex domain. We have shown significant improvement over the baseline both in terms of classification accuracy and number of parameters. Furthermore, we have shown that using smaller number of parameters ComplexNet gives similar accuracy to the baseline model on the popular radio frequency data, RadioML. In future, we would like to apply our proposed framework on other complex-valued datasets and also explore the representation power of the filters using the proposed convolutional operator.

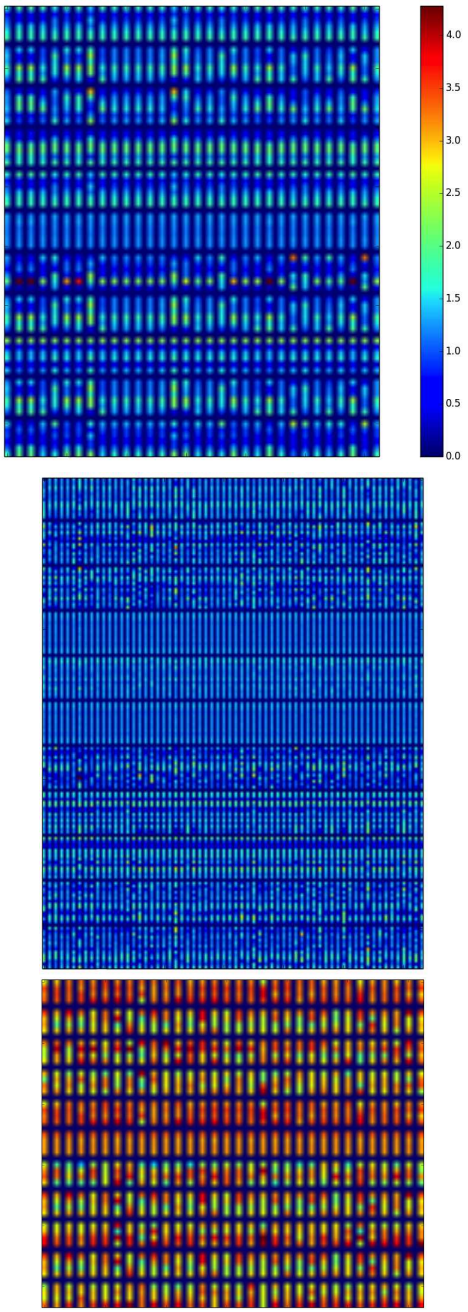

Figure 8: *(Let-Right, Top-Bottom)* Representative filter outputs after the first, second, third convolution layers of ComplexNet on RadioML data

5. Acknowledgement

This research was supported, in part, by Berkeley Deep Drive, DARPA, and US Government fund through Etegent Technologies on Low-Shot Detection in Remote Sensing Imagery. The views, opinions and/or findings expressed are those of the author and should not be interpreted as representing the official views or policies of the Department of Defense or the U.S. Government.

References

[1] William M Boothby. *An introduction to differentiable manifolds and Riemannian geometry*, volume 120. Academic press, 1986.

[2] Kerstin Bunte, Frank-Michael Schleif, and Michael Biehl. Adaptive learning for complex-valued data. In *ESANN*. Citeseer, 2012.

[3] Rudrasis Chakraborty, Monami Banerjee, and Baba C Vemuri. H-cnns: Convolutional neural networks for riemannian homogeneous spaces. *arXiv preprint arXiv:1805.05487*, 2018.

[4] Rudrasis Chakraborty, Jose Bouza, Jonathan Manton, and Baba C Vemuri. Manifoldnet: A deep network framework for manifold-valued data. *arXiv preprint arXiv:1809.06211*, 2018.

[5] Taco Cohen and Max Welling. Group equivariant convolutional networks. In *International conference on machine learning*, pages 2990–2999, 2016.

[6] Taco S Cohen, Mario Geiger, Jonas Köhler, and Max Welling. Spherical CNNs. *arXiv preprint arXiv:1801.10130*, 2018.

[7] Sander Dieleman, Kyle W. Willett, and Joni Dambre. Rotation-invariant convolutional neural networks for galaxy morphology prediction. *Monthly Notices of the Royal Astronomical Society*, 2015.

[8] David Steven Dummit and Richard M Foote. *Abstract algebra*, volume 3. Wiley Hoboken, 2004.

[9] Carlos Esteves, Christine Allen-Blanchette, Xiaowei Zhou, and Kostas Daniilidis. Polar Transformer Networks. *arXiv preprint arXiv:1709.01889*, 2017.

[10] William T. Freeman and Edward H Adelson. The design and use of steerable filters. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (9):891–906, 1991.

[11] Sigurdur Helgason. *Differential geometry and symmetric spaces*, volume 12. Academic press, 1962.

[12] Eric R Keydel, Shung Wu Lee, and John T Moore. Mstar extended operating conditions: A tutorial. In *Algorithms for Synthetic Aperture Radar Imagery III*, volume 2757, pages 228–243. International Society for Optics and Photonics, 1996.

[13] Risi Kondor and Shubhendu Trivedi. On the generalization of equivariance and convolution in neural networks to the action of compact groups. *arXiv preprint arXiv:1802.03690*, 2018.

[14] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. ImageNet Classification with Deep Convolutional Neural Networks. *Advances In Neural Information Processing Systems*, 2012.

[15] Michael Maire, Takuya Narihira, and Stella X Yu. Affinity cnn: Learning pixel-centric pairwise relations for figure/ground embedding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 174–182, 2016.

[16] Stéphane Mallat. Understanding Deep Convolutional Networks. *Philosophical Transactions A*, 374:20150203, 2016.

[17] Maurice Fréchet. Les éléments aléatoires de nature quelconque dans un espace distancié. *Annales de l'I. H. P.,*, 10(4):215–310, 1948.

[18] Timothy J O'Shea, Johnathan Corgan, and T. Charles Clancy. Convolutional radio modulation recognition networks. *arXiv preprint arXiv:1602.04105*, 2016.

[19] Timothy J O'Shea and Nathan West. Radio machine learning dataset generation with gnu radio. *Proceedings of the 6th GNU Radio Conference*, 2016.

[20] Chiheb Trabelsi, Olexa Bilaniuk, Ying Zhang, Dmitriy Serdyuk, Sandeep Subramanian, João Felipe Santos, Soroush Mehri, Negar Rostamzadeh, Yoshua Bengio, and Christopher J Pal. Deep complex networks. *arXiv preprint arXiv:1705.09792*, 2017.

[21] Daniel E Worrall, Stephan J Garbin, Daniyar Turmukhambetov, and Gabriel J Brostow. Harmonic networks: Deep translation and rotation equivariance. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, volume 2, 2017.

[22] Stella Yu. Angular embedding: A robust quadratic criterion. *IEEE transactions on pattern analysis and machine intelligence*, 34(1):158–173, 2012.