SurReal: Fréchet 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 a novel fully connected layer operator using the distance to the wFM, with natural equivariant properties to non-zero scaling and planar rotation for the former and invariance properties for the latter.

Compared to the baseline approach of learning real-valued neural network models on the two-channel real-valued representation of complex-valued data, our method achieves surreal performance on two publicly available complex-valued datasets: MSTAR on SAR images and RadioML on radio frequency signals. On MSTAR, at 8% *of the baseline model size and with fewer than* $45,000$ *parameters, our model improves the target classification accuracy from* 94% *to* 98% *on this highly imbalanced dataset. On RadioML, our model achieves comparable RF modulation classification accuracy at* 10% *of the baseline model size.*

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) images, 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 the size of contrast or geometrical shapes. Even for real-valued images, their complex-valued representations could be more successful for many pattern recognition tasks; the most notable examples are the Fourier spectrum and spectrum-based computer vision techniques ranging from steerable filters [10] to spectral graph embedding [16, 24].

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 geometry of complex-valued data. For example, in MR and SAR images, the pixel intensity value could be subject to complex-valued scaling. One way to get around such an ambiguity is to train a model with data augmentation [15, 7, 22], 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 rotation in the complex plane.

We treat each complex-valued data sample as a field in the space of complex numbers, which is a special non-Euclidean space. This perspective allows us to develop novel concepts for both convolution and fully connected layer functions that achieve equivariance and invariance to complex-valued scaling.

A major hurdle in extending convolution from the Euclidean space to a non-Euclidean space is the lack of a vector space structure. In the Euclidean space, there exists a translation to go from one point to another, and convolution is equivariant to translation. In a non-Euclidean space such as a sphere, a point undergoing translation may no longer remain in that space, hence translation equivariance is no longer meaningful. What is essential and common between a non-Euclidean space and the Euclidean space is that, there is a group that transitively acts in the space. For example, there is a rotation, *instead of translation*, to go from one point to another on a sphere. Extending convolution to a non-Euclidean space should consider equivariance to some transitive action group specific to that space.

Note that such a manifold view applies to both the domain and the range of the data space. To extend deep learning to complex-valued images or signals, we take the manifold perspective towards the *range* space of the data.

There is a long line of works that define convolution in a non-Euclidean space by treating each data sample as a

function in that space [23, 5, 6, 9, 3, 14].

Our key insight is to represent a complex number by its polar form, such that the general group that acts in this space is the product of planar rotation and non-zero scaling. This representation turns the complex plane into a particular Riemannian manifold. We want to define convolution that is equivariant to the action of this product group in that space.

When a sample is a field on a Riemannian manifold,

- Convolution defined by weighted Fréchet mean (wFM) [18] is equivariant to the group that naturally acts on that manifold [4].
- Non-linear activation functions such as ReLU may not be needed, since wFM is a non-linear contraction mapping [17] analogous to ReLU or sigmoid.
- Taking the Riemannian geometric point of view, we could also use tangent ReLU for better accuracy.
- We further propose a distance transform as a fully-connected layer operator that is invariant to complex scaling. It takes complex-valued responses at a previous layer to the real domain, where all kinds of standard CNN functions can be subsequently used.

A neural network equipped with our wFM filtering and distance transform on complex-valued data has a group invariant property similar to the standard CNN on real-valued data. Existing complex-valued CNNs tend to extend the real-valued counterpart to the complex domain based on the form of functions [2, 21], e.g. convolution or batch normalization. None of complex-valued CNNs are derived by studying the desired property of functions, such as equivariance or linearity. Our complex-valued CNN is composed of layer functions with all the desired properties and is a theoretically justified analog of the real-valued CNN.

On the SAR image dataset MSTAR, compared to the baseline of a real-valued CNN acting on the two-channel real representation of complex-valued data and reaching 94% accuracy, our complex-valued CNN acting directly on the complex-valued data (i.e., also without any preprocessing) achieves 98% target classification accuracy with only 8% of parameters. Likewise, on the radio frequency signal dataset RadioML, our method achieves comparable modulation mode classification (a harder task than target recognition) performance with fewer parameters.

To summarize, we make two major contributions.

1. We propose novel complex-valued CNNs with theoretically proven equivariance and invariance properties.
2. We provide *sur-real* (pun intended) experimental validation of our method on complex-valued data classification tasks, demonstrating significant performance gain at a fraction of the baseline model size.

These results demonstrate significant benefits of designing new CNN layer functions with desirable properties on the complex plane as opposed to applying the standard CNN to the 2D Euclidean embedding of complex numbers.

2. Our Complex-Valued CNN Theory

We first present the geometry of the manifold of complex numbers and then develop complex-valued convolutional neural network (CNN) on that manifold.

Space of complex numbers. Let $\mathbf{R}$ denote the set of real numbers. All the complex number elements assume the form $a + ib$, where $i = \sqrt{-1}$, $a, b \in \mathbf{R}$, and lie on a a Riemannian manifold [1] denoted by $\mathbf{C}$. The distance induced by the canonical Riemannian metric is:

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

We identify $\mathbf{C}$ with the polar form of complex numbers.

Definition 1. *We identify each complex number, $a+ib$, with its polar form, $r\exp(i\theta)$, where r and θ are the absolute value (*abs*) or magnitude and argument (*$\arg$*) or phase of $a + ib$. Here $\theta \in [-\pi, \pi]$. Hence, we can identify $\mathbf{C}$ as $\mathbf{R}^+ \times \mathbf{SO}(2)$, where $\mathbf{R}^+$ is the set of positive numbers, and $\mathbf{SO}(2)$ is the manifold of planar rotations. Let $F : \mathbf{C} \to \mathbf{R}^+ \times \mathbf{SO}(2)$ be the mapping from the complex plane to the product manifold $\mathbf{R}^+ \times \mathbf{SO}(2)$:*

$$\begin{aligned} a + ib &\overset{F}{\mapsto} (r, R(\theta)), \\ r &= \mathrm{abs}(a + ib) = \sqrt{a^2 + b^2} \\ \theta &= \arg(a + ib) = \mathrm{atan2}(b, a) \\ R(\theta) &= \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix}. \end{aligned}$$

Note that F is bijective.

Manifold distance between complex numbers. The geodesic distance on this manifold is the Euclidean distance induced from Eq. (1) in the tangent space. Given $\mathbf{z}_1, \mathbf{z}_2 \in \mathbf{C}$, let $(r_1, R_1) = F(\mathbf{z}_1)$ and $(r_2, R_2) = F(\mathbf{z}_2)$. While the Euclidean distance between two complex numbers is Eq. (1), their manifold distance $\mathbf{R}^+ \times \mathbf{SO}(2)$ is:

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

where logm is the matrix logarithm. Note that, for $A = R(\theta) \in \mathbf{SO}(2)$, we choose $\mathrm{logm}(A)$ to be $\theta \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$.

Transitive actions and isometries. $\mathbf{C}$ is in fact a *homogenous Riemannian manifold* [11], a topological space on which there is a group of actions acts transitively [8].

Definition 2. *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)*** $(gh).X = g.(h.X)$. *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. *Group* $G := \{\mathbf{R} \setminus \{0\}\} \times \mathbf{SO}(2)$ *transitively acts on* $\mathbf{C}$ *and the action is given by* $((r,R),(r_g,R_g)) \mapsto \left(r_g^2 r, R_g R\right)$.

It is straightforward to verify that group G transitively acts on $\mathbf{C}$. We show that G is the set of isometries on $\mathbf{C}$.

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

The proof follows from the definitions of d and g:

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

Having defined our manifold range space for complex numbers, we focus on extending two key properties, *equivariance* of a convolution operator and *invariance* of a CNN, from real-valued CNNs to complex-valued CNNs.

Equivariance property of convolution. In the Euclidean space $\mathbf{R}^n$, the convolution operator is equivariant to translation: Given the kernel of convolution, if the input is translated by $\mathbf{t}$, the output would also be translated by $\mathbf{t}$. This property enables weight sharing across the entire spatial domain of an image. The group of translations is the group of isometries for $\mathbf{R}^n$, and it transitively acts on $\mathbf{R}^n$.

We extend these concepts to our complex number manifold $\mathbf{C}$. Our $G = \{\mathbf{R} \setminus \{0\}\} \times \mathbf{SO}(2)$ transitively acts on $\mathbf{C}$ and is the group of isometries. In order to generalize the Euclidean convolution operator on $\mathbf{C}$, we need to define an operator on $\mathbf{C}$ which is equivariant to the action of G.

CNNs on manifold valued data have recently been explored in [4], where convolution is defined on manifold $\mathcal{M}$ and equivariant to the group G that acts on $\mathcal{M}$. In our case, manifold $\mathcal{M} = \mathbf{C}$ and action group $G = \{\mathbf{R}\setminus\{0\}\} \times \mathbf{SO}(2)$.

Convolution as manifold Fréchet mean filtering. Given K points on our manifold $\mathbf{C}$: $\{\mathbf{z}_i\}_{i=1}^K \subset \mathbf{C}$, and K nonnegative weights $\{w_i\}_{i=1}^K \subset (0,1]$ with $\sum_i w_i = 1$, the *weighted Fréchet mean (FM)* (wFM) is defined as [18]:

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

where d is the distance defined in Eq. (2). Unlike the standard Euclidean convolution which *evaluates* the weighted data mean given the filter weights, the manifold convolution wFM *solves* the data mean that minimizes the weighted variance. There is no closed-form solution to wFM; however, there is a provably convergent $K-$step iterative solution [4].

While our filter response $\mathsf{wFM}(\{\mathbf{z}_i\}, \{w_i\}) \in \mathbf{C}$ is complex-valued, a minimizing argument to Eq. (3), the filter weights $\{w_i\}$ themselves are real-valued. They are learned through stochastic gradient descent, subject to additional normalization and convexity constraints on $\{w_i\}$.

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

The equivariance property of convolution follows from the isometry in Prop. (2). Fig (1) illustrates the equivariance of wFM with respect to planar rotation and scaling.

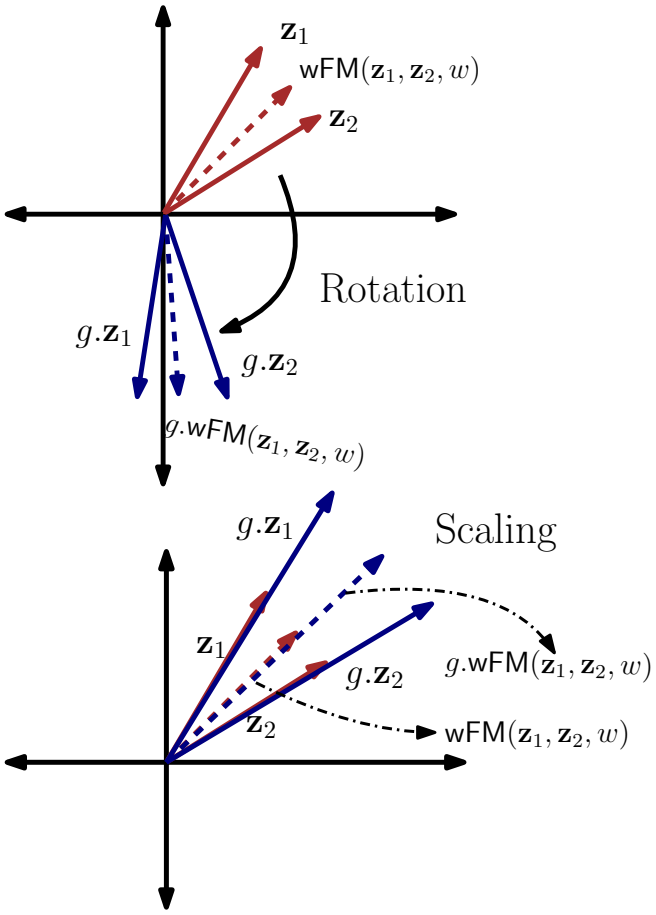

Figure 1: Equivariance of weighted Fréchet mean filtering with respect to rotation and scaling in the complex plane.

Manifold vs. Euclidean convolution. Convolution is often written as $\sum_i w_i x_i$, where $\{w_i\}$ is the filter and $\{x_i\}$ is the signal. With our 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 defined in Eq. (3). The convexity constraint is to ensure that the resultant stays on the manifold. Therefore, wFM as a convolution operator on the manifold might appear rather arbitrary at first glance, it is an obvious choice if we regard the standard convolution as the minimizer of the weighted variance in the Euclidean space.

Next we turn to nonlinear activation functions. Our wFM is non-linear and *contractive* [4], it thus performs not only convolution but also nonlinear activation to a certain extent. Nevertheless, we extend ReLU in the Euclidean space to a manifold in a principled manner.

ReLU on the manifold: tReLU. The tangent space of a manifold could be regarded as a local Euclidean approximation of the manifold, and a pair of transformations, logarithmic and exponential maps, establish the correspondence between the manifold and the tangent space.

Our tReLU is a function from $\mathbf{C}$ to $\mathbf{C}$, just like the Euclidean ReLU from $\mathbf{R}^n$ to $\mathbf{R}^n$, but it is composed of three steps: **1)** Apply logarithmic maps to go from a point in $\mathbf{C}$ to a point in its tangent space; **2)** Apply the Euclidean ReLU in the tangent space; **3)** Apply exponential maps to come back to $\mathbf{C}$ from the tangent space.

$$(r, R) \overset{\text{tReLU}}{\mapsto} \\ (\ \exp(\text{ReLU}(\log(r))), \text{expm}(\text{ReLU}(\text{logm}(R)))\ ) \quad (4)$$

where expm is the matrix exponential operator. Our manifold perspective leads to a non-trivial extension of ReLU, partitioning the complex plane by r and θ into four scenarios, e.g., those with $r<1$ would be rectified to $r=1$.

Invariance property of a CNN classifier. For classification tasks, having equivariance of convolution and range compression of nonlinear activation functions are not enough; we need the final representation of a CNN invariant to within-class feature variations.

In a standard Euclidean CNN classifier, the entire network is invariant to the action of translations, achieved by the fully connected (FC) layer. Likewise, we develop a FC function on $\mathbf{C}$ that is invariant to the action of G.

Distance transform as an invariant FC layer. Since our distance d is shown invariant to G, we propose the distance of each point in a set to their weighted Fréchet mean, which is equivariant to G, as a new FC function on $\mathbf{C}$.

Consider turning an m-channel s-dimensional feature representation, $\{\mathbf{t}_i\}_{i=1}^m \subset \mathbf{C}$, into a single FC feature u of m dimensions. Each input channel $\mathbf{t}_i$ contains s elements (in any matrix shape) and is treated as an s-dimensional feature vector. Our distance transform first computes the wFM of m input features and then turns input channel i into a single scalar u_i as its distance to the mean:

$$\mathbf{m} = \mathsf{wFM}(\{\mathbf{t}_i\}, \{v_i\}) \quad (5)$$

$$u_i = d(\mathbf{t}_i, \mathbf{m}), \quad (6)$$

The m filter weights v_i are learned per FC output channel, and there could be multiple output channels in the FC layer.

Proposition 4. *The above distance transform, defined as the distance to the wFM, is invariant to the action of G.*

The proof follows from Propositions 2 and 3:

$$\begin{aligned} & d(g.\mathbf{t}_i, \mathsf{wFM}(g.\{\mathbf{t}_i\}, \{v_i\})) \\ =& d(g.\mathbf{t}_i, g.\mathsf{wFM}(\{\mathbf{t}_i\}, \{v_i\})) && \text{equivariance of wFM} \\ =& d(\mathbf{t}_i, \mathsf{wFM}(\{\mathbf{t}_i\}, \{v_i\})) && \text{invariance of distance.} \quad \square \end{aligned}$$

With our distance transform, complex-valued intermediate feature representations are turned into real values, upon which we can apply any of the standard layer functions in the real domain, such as softmax to the last layer of c channels for c-way classification.

Complex-valued neural network. With these new convolution, nonlinear activation, and FC layer functions, we can construct a complex-valued CNN which is invariant to the action of G. Fig. (2) illustrates a possible CNN architecture. Alg. (1) presents a CNN work-flow with two convolution layers and one FC layer.

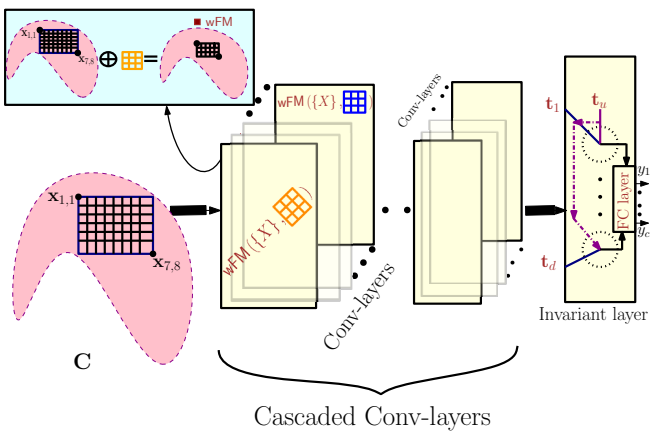

Figure 2: Sample architecture of our complex-valued CNN classifier that is invariant to G. It consists of our newly proposed wFM convolution, tReLU nonlinear activation, and FC distance transform layer functions, together achieving invariance to complex-valued scaling in the range space.

Algorithm 1: Workflow of our sample CNN classifier with 2 convolution layers and 1 FC layer.

function COMPLEX-VALUED CNN 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

3. Experimental Results

We conduct our experiments on two publicly available complex-valued datasets: MSTAR [12] and RadioML [19, 20]. MSTAR contains complex-valued 2D SAR images, and RadioML contains complex-valued 1D RF signals.

3.1. MSTAR Experiments

MSTAR dataset. It consists of X-band SAR image chips with 0.3m $\times$ 0.3m resolution of 10 target classes such as infantry combat vehicle (BMP2) and armored personnel carrier BTR70. The number of instances per class varies greatly from 429 to 6694. We crop 100×100 center regions from each image without other preprocessing (Fig. (3)).

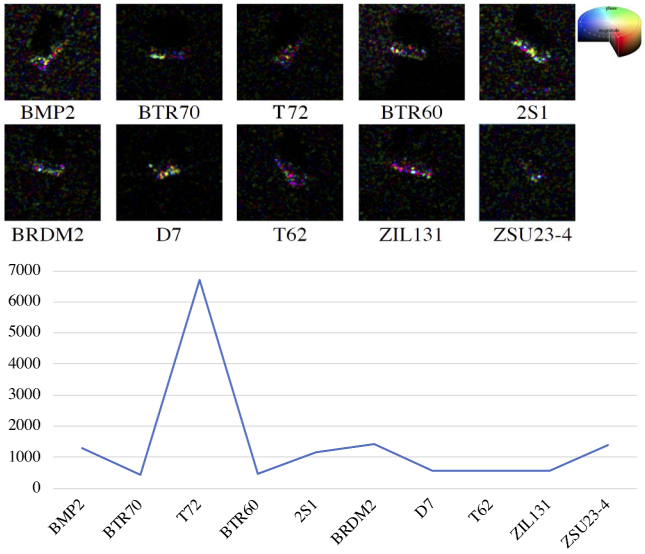

Figure 3: MSTAR has 10 imbalanced target classes, a sample image per class shown on the top and the total number of images per class shown in the bottom. We use the HSV color map for visualizing the phase of a complex-valued image. SAR images are noisy with a large intensity range.

MSTAR baselines. We use the real-valued CNN model in Fig. (8) and consider 4 possible representations of complex-valued inputs as real-valued data. Let $\mathbf{z} = a + ib = re^{j\theta}$.

1. (a, b): Treat a 1-channel complex-valued image as a 2-channel real-valued image, with real and imaginary components in two separate channels.
2. r: Take only the absolute value of a complex-valued image to make a 1-channel real-valued image, with the phase of complex numbers ignored.
3. (a, b, r): Take both the real, imaginary, and magnitude of a complex-valued image to make a 3-channel real-valued image.
4. (r, θ): Take the magnitude and phase of a complex-valued image to make a 2-channel real-valued image.

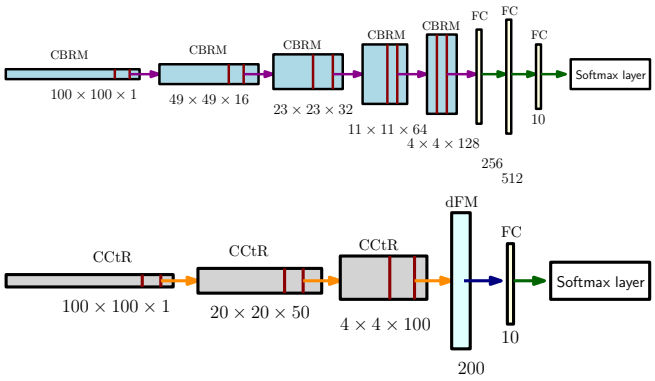

Figure 4: Real-valued CNN baseline model (top) and our complex-valued model (bottom) for MSTAR. CBRM denotes Conv, Batch-Normalization, ReLU, Pooling. CCtR denotes Complex Convolution, tangent ReLU.

We perform a 30-70 random train-test split and report the average classification accuracy over 10 runs.

Our CNN model. 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 100.

MSTAR results. Table (1) shows the confusion matrix and the overall classification accuracy for each of the four real-valued CNN baseline and our complex-valued CNN model. Ours has a 3.6% accuracy gain over the best baseline.

This performance gain has to come from the group equivariant property of our convolution and the group invariant property of our CNN classifier. The group that acts on the complex numbers is $\mathbf{R} \setminus \{0\} \times \mathsf{SO}(2)$. Our equivariance and invariance properties guarantee that our learned CNN is invariant to scaling and planar rotations, unlike any standard real-valued CNN architecture. Table (1) also suggests that our learned CNN is more robust to the imbalanced training data. For example, on the smallest class 'BTR70' with test set size 429, our model correctly classifies 406 samples while the baseline correctly classifies only 172 samples.

Among the real-valued baselines, just the magnitude r alone gives a better classification accuracy than the two-channel real-valued representation (a, b). Their combination (a, b, r) achieves a classification accuracy of 96.87%, with 2% improvement over the magnitude only representation of r. The polar representation (r, θ) is better than the two-channel real-imaginary representation (a, b), but is in fact worse than the magnitude r only representation. A natural question is whether phase information is useful at all.

How useful is phase alone? We remove any useful in-

(a, b) : 89.77%

84.5	2.1	0.9	11.7		0.6		0.2	0.1	
0.2	**78.3**		21.2				0.2		
0.5		**94.2**	0.9	0.2	0.1		3.8		0.2
	0.7		**99.3**						
0.8	1.6	0.4	4.6	**81.7**	6.2		4.6	0.1	0.1
0.1			5.3	0.1	**94.1**			0.4	
		4.2	0.3		1.2	**88.5**	2.1	1.9	1.7
		7.7	4.4	0.2	0.2		**87.6**		
		4.2	1.2	0.5	0.5		0.5	**93.0**	
0.1		8.9	2.4		8.2	0.6	3.1	0.4	**76.4**

r : 94.46%

95.3	4.0	0.5		0.2					
	98.6	0.7		0.7					
0.4	0.1	**99.2**			0.1		0.1		0.1
0.9	65.4	4.7	**22.2**	1.8	0.4		4.7		
0.1	3.4	1.1		**94.0**	1.0		0.1		0.3
2.9	0.6	0.3		0.4	**94.4**	0.1	0.1	1.0	0.3
		0.2				**98.8**		0.2	0.9
		21.5		2.4			**75.5**	0.2	0.3
		3.0		1.0		0.3		**94.9**	0.7
		0.6				0.2			**99.1**

(a, b, r) : 96.87%

97.0	0.1	0.9	0.5	0.5	1.0			0.1	
3.5	**90.4**		4.4	0.9	0.7				
0.1		**98.5**		0.1	0.1		0.2	0.1	0.9
1.6	0.2	0.2	**96.9**	0.2	0.7	0.2			
0.1		0.3	0.3	**97.3**	1.3		0.2	0.4	0.1
0.2					**99.4**	0.1		0.9	0.1
0.2						**99.0**		0.2	0.7
0.2		7.9		0.3		0.3	**86.7**	3.3	1.2
				0.2	0.2	2.3		**97.4**	
0.3		0.6		0.4	0.1	5.8		0.1	**92.8**

(r, θ) : 93.51%

91.7	0.2	1.8		4.4	0.5	0.1	1.2	0.2	
5.1	**86.2**	0.2	0.7	7.7					
0.2		**96.8**		0.5			1.9	0.3	0.3
9.5	13.5		**56.1**	16.9	1.8	0.7	1.3	0.2	
0.1	0.1	1.3		**96.6**	0.1		1.1	0.6	0.1
0.1	0.1	0.6		3.3	**94.1**	0.1		1.3	0.4
						99.7		0.2	0.2
		11.0		0.3			**86.0**	2.4	0.2
						1.0	0.2	**98.6**	0.2
		6.7		0.1	0.1	1.9	0.6	1.4	**89.2**

z : 98.16%

97.8	0.1	1.9		0.2	0.1				
1.4	**97.4**	0.2	0.7	0.2					
0.4		**99.0**		0.1	0.1				0.4
4.2	1.8	1.1	**90.2**	1.6	1.1				
	0.2	1.8		**96.4**	1.0				0.6
		0.4		0.1	**98.9**	0.1			0.5
						10			
		4.9				0.2	**94.4**	0.5	
		1.2				0.5		**98.3**	
		0.9		0.1	0.1				**98.9**

Table 1: Confusion matrices for 4 real-valued baselines and our complex-valued CNN. The method and the overall accuracy is listed at the top left corner of each table. The order of categories is the same as that in Fig. 3.

formation in the magnitude by normalizing each complex number to norm 1. On the normalized complex numbers, Table (2) shows the classification confusion matrix for the baseline (a, b) CNN model and our model. The real-valued CNN achieves an overall accuracy of 45.98%, with all the

(a, b) : 45.98%

		100							
		100							
		100							
		100							
		100							
		100							
		100							
		100							
		100							

z :97.00%

95.6	0.3	2.9	0.8	0.2	0.2				
2.6	**94.4**	0.2	2.3	0.5					
0.6		**97.9**		0.4	0.3			0.2	0.4
2.9	2.0	1.6	**90.7**	2.0	0.9				
	0.3	1.5	0.3	**94.8**	2.6		0.1	0.3	0.3
		0.6		0.5	**98.4**	0.1		0.1	0.4
						99.7		0.3	1.7
		6.8		0.7	0.2		**91.1**	0.9	0.3
	0.2	0.7				0.3		**98.8**	
		1.5		0.4	0.3	0.1	0.1	0.1	**97.6**

Table 2: Confusion matrices for the baseline model (a, b) (top) and our model (bottom) applied to normalized complex numbers. Same convention as Table 1. With an overall accuracy of 97% over the baseline accuracy 46%, our complex-valued CNN brings significant discrimination power out of the phase information alone.

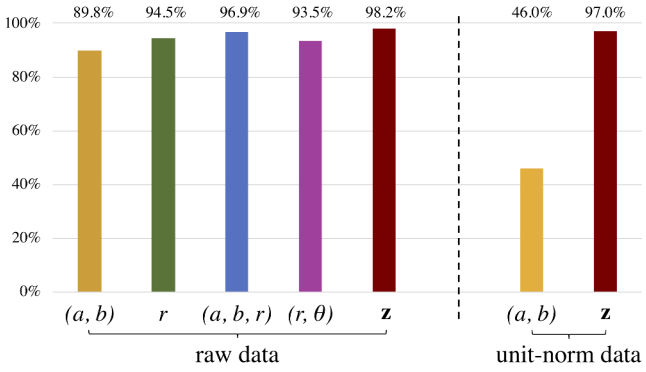

Figure 5: MSTAR classification accuracies by real-valued baseline CNNs and our complex-valued CNN, with raw and normalized complex number inputs.

test set classified as the largest class which consists of 45.98% samples of the entire dataset. That is, the real-valued CNN is completely confused by the phase and unable to tease apart different classes. On the other hand, our model gives a surprisingly high accuracy of 97%, only 1% less than our result on the raw complex numbers which contains the class-discriminative magnitude.

Fig. (5) compares the classification accuracies in different settings. The stark contrast in real- and complex-valued CNNs to phase data alone demonstrates not only the effectiveness of our complex-valued CNN due to its invariance to G, but also the richness of the phase information alone.

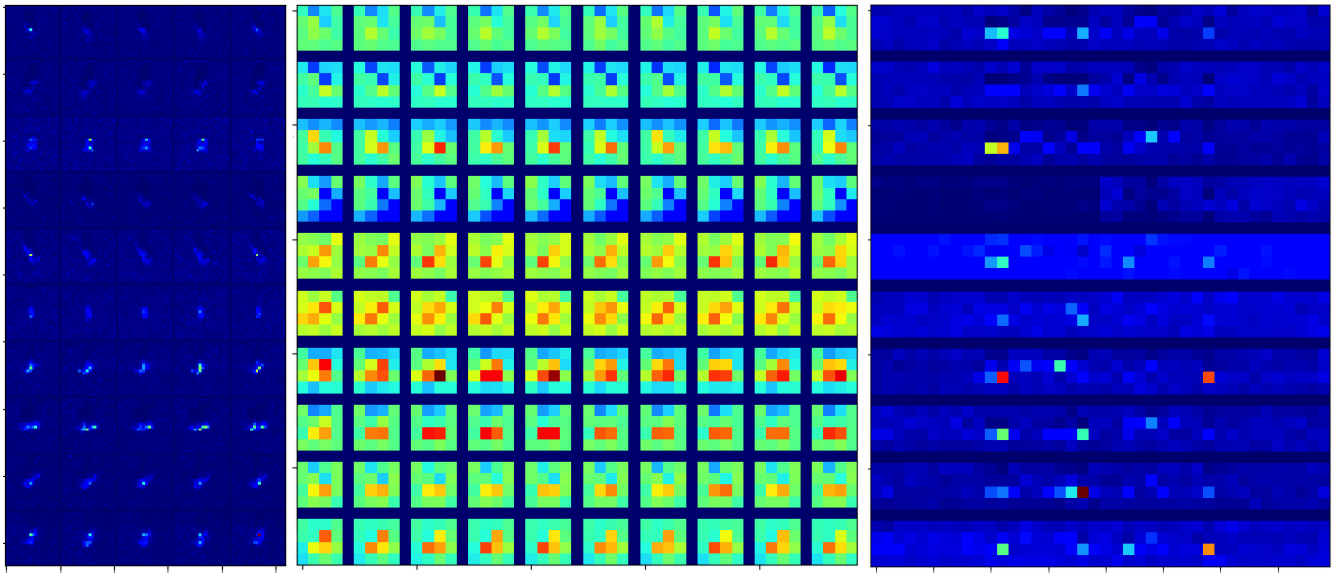

Figure 6: Sample MSTAR filter responses of our model after the first, second and third conv layer. Each row corresponds to the same image; each column represents a particular channel's absolute valued response.

CNN model	domain representation	# parameters
real	(a, b)	$530,170$
real	r	$530,026$
real	(a, b, r)	$530,314$
real	(r, θ)	$530,170$
complex	$\mathbf{z}$	$\mathbf{44,826}$

Table 3: CNN model size comparison. Our complex-valued CNN is 8% of the baseline real-valued CNN model size.

Our complex-valued CNN is better and leaner. Table (3) lists the total number of parameters used in each CNN model. As our complex-valued CNN captures the natural equivariance and invariance in the non-Euclidean complex number range space, which standard CNNs fail to do, our model achieves a higher accuracy with a significant (more than 90%) parameter reduction.

CNN visualization. Fig. (6) shows examples of filter responses at three convolution layers on the representative images in Fig. (3). The first convolution layer produces basically blurred versions of the input image. From the second convolution layer onward, the filter response patterns grow more divergent for different classes. While we show one sample output from each class, the patterns within each class are similar. For classes 'D7', 'T62', 'ZIL131', the filter responses are higher than the other classes. Furthermore, the last convolution layer shows significantly different patterns between different classes.

3.2. RadioML Experiments

RadioML dataset. RF modulation operates on both discrete binary alphabets (digital modulations) and continuous alphabets (analog modulations). Over each modem the known data is modulated and then exposed to channel ef-

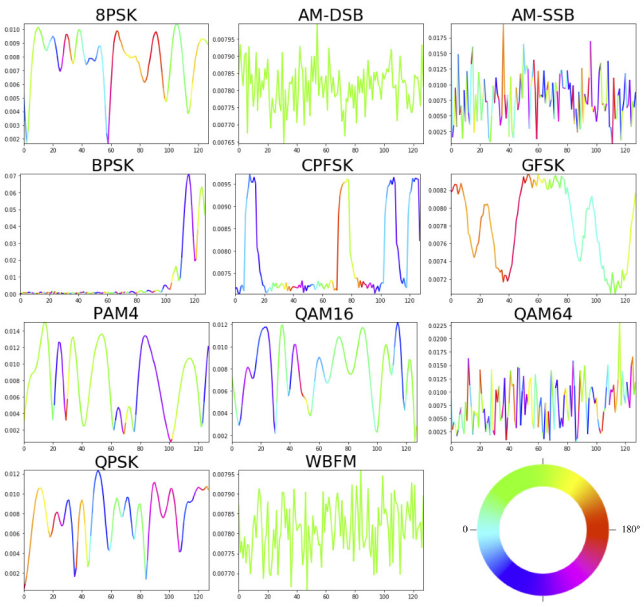

Figure 7: RadioML data samples. We plot one sample per class at SNR 18. We use the HSV colormap to encode and visualize the phase of complex valued 1D signals.

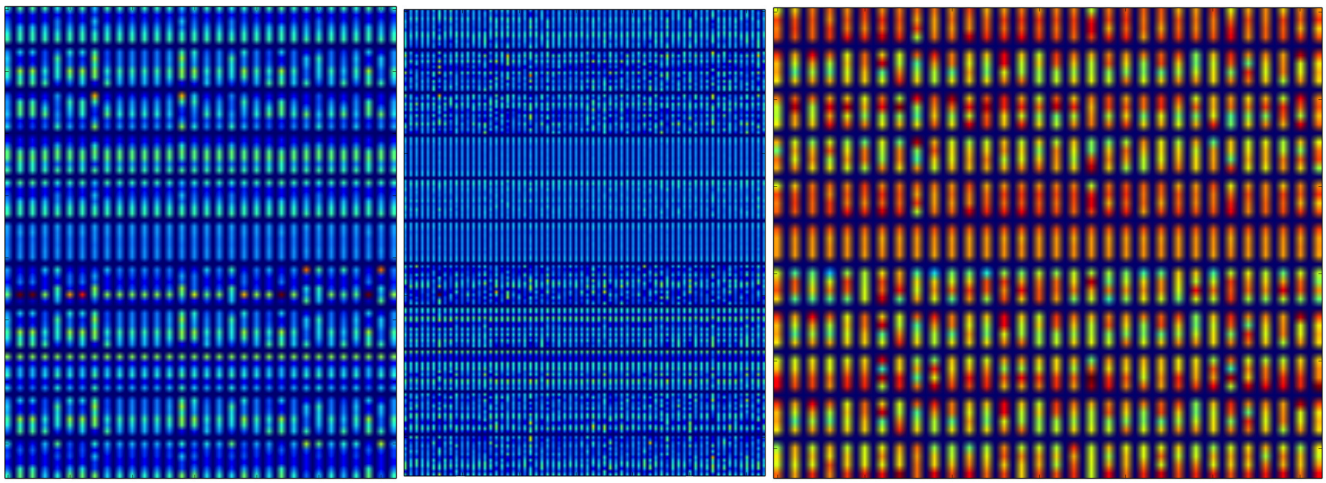

Figure 8: Representative filter outputs after the first, second, third convolutional layers (absolute valued responses) of our complex-valued network on the RadioML data. Same convention as Fig. 6.

fects using GNU Radio. It is then segmented into short-time windows in a fashion similar to how a continuous acoustic voice signal is typically windowed for voice recognition tasks. Fig. (7) visualizes these 1D complex-valued time series as colored lines. There are $220,000$ samples in RadioML [19, 20]. We use a 50-50 train-test split and 10 random runs as in our MSTAR experiments.

RadioML baseline. It consists of two convolutional and two fully connected layers as used in [19]. The convolution kernel is of size 3 with 256 and 80 channels respectively. Each convolutional layer is followed by ReLU and dropout layers. This network has $2,830,491$ parameters.

Our RadioML CNN model. It has two complex convolutional layers of stride 5, kernel sizes 7 and 5, the numbers of channels 64 and 128, followed by an invariant distance transform layer and a final softmax layer for classification. Fig. 9 shows both the real-valued baseline CNN and our complex-valued CNN architectures. We use ADAM optimizer [13] with learning rate 0.05 and mini-batch size 500.

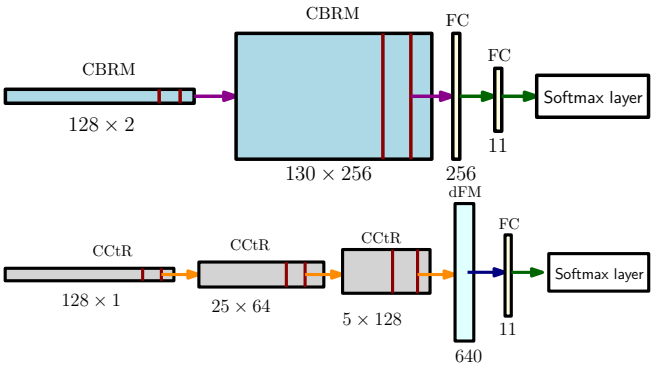

Figure 9: Real-valued CNN model and our complex-valued model for RadioML. CBRM denotes Convolution, Batch-Normalization, ReLU, and Pooling. CCtR denotes our Complex-valued Convolution and tangent ReLU, and dFM our distance transform with respect to the Fréchet mean.

Our complex-valued CNN has only $299,117$ parameters, i.e., roughly 10% of the baseline model, yet it can achieve test accuracy 70.23%, on par with 70.68% of the baseline real-valued CNN model. This lean model result is consistent with our MSTAR experiments. Fig. (8) also shows that discriminative filter response patterns emerge quickly from various smoothing effects of convolutional layers.

4. Summary

We take a manifold view on complex-valued data and present a novel CNN theory. Our convolution from Fréchet mean filtering is equivariant and our distance transform is invariant to complex-valued scaling, an inherent ambiguity in the complex value range space.

Our experiments on MSTAR and RadioML demonstrate that our complex-valued CNN classifiers can deliver better accuracies with a surreal leaner CNN model, at a fraction of the real-valued CNN model size.

By representing a complex number as a point on a manifold instead of two independent real-valued data points, our model is more robust to imbalanced classification and far more powerful at discovering discriminative information in the phase data alone.

Acknowledgements. This research was supported, in part, by Berkeley Deep Drive and DARPA. 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] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[14] 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.

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

[16] 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.

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

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

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

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

[21] 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.

[22] Jiayun Wang, Patrick Virtue, and Stella X Yu. Joint embedding and classification for sar target recognition. *arXiv preprint arXiv:1712.01511*, 2017.

[23] 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.

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