

# SurReal: Complex-Valued Learning as Principled Transformations on a Scaling and Rotation Manifold

Rudrasis Chakraborty<sup>1</sup>, Yifei Xing<sup>1</sup>, and Stella X. Yu<sup>1</sup>

**Abstract**—Complex-valued data are ubiquitous in signal and image processing applications, and complex-valued representations in deep learning have appealing theoretical properties. While these aspects have long been recognized, complex-valued deep learning continues to lag far behind its real-valued counterpart. We propose a principled geometric approach to complex-valued deep learning. Complex-valued data could often be subject to arbitrary complex-valued scaling; as a result, real and imaginary components could covary. Instead of treating complex values as two independent channels of real values, we recognize their underlying geometry: we model the space of complex numbers as a product manifold of nonzero scaling and planar rotations. Arbitrary complex-valued scaling naturally becomes a group of transitive actions on this manifold. We propose to extend the property instead of the form of real-valued functions to the complex domain. We define convolution as the weighted Fréchet mean on the manifold that is equivariant to the group of scaling/rotation actions and define distance transform on the manifold that is invariant to the action group. The manifold perspective also allows us to define nonlinear activation functions, such as tangent ReLU and  $G$ -transport, as well as residual connections on the manifold-valued data. We dub our model SurReal, as our experiments on MSTAR and RadioML deliver high performance with only a fractional size of real- and complex-valued baseline models.

**Index Terms**—Complex value, equivariance, Fréchet mean, invariance, Riemannian manifold.

## I. INTRODUCTION

WHILE deep learning has been widely successful in computer vision and machine learning [3], [25], [32]–[34], most techniques are only applicable to data that lie in a vector space. How to handle manifold-valued data and incorporate non-Euclidean geometry into deep learning has become an active topic of research [6], [12]–[14], [17], [21].

We are interested in extending deep learning to complex-valued data, e.g., synthetic aperture radar (SAR) images in remote sensing, magnetic resonance (MR) images in medical imaging, or radio frequency (RF) signals in electrical engineering. For such naturally complex-valued data, both the size (or magnitude) and the phase of a complex-valued measurement contain useful information. For example, in SAR

images, the magnitude encodes the amount of energy, whereas the phase variation indicates the object material and shape boundaries.

Complex-valued data could also arise from the more informative complex-valued representation of naturally real-valued data. The most notable examples are the Fourier spectrum and spectrum-based computer vision techniques ranging from steerable filters [23] to spectral graph embedding [35], [54].

The most common complex-valued deep learning approach is to simply apply real-valued deep learning methodology to the two-channel representation of complex-valued data  $z = x + i \cdot y$  (where  $i$  denotes the imaginary unit): the real component  $x$  and the imaginary component  $y$  are regarded as the independent channels of the input.

However, the independence assumption between the real and imaginary components does not hold in general. For instance, in MR and SAR images, the pixel intensity value could be subject to arbitrary scaling by complex number  $s = m e^{j\theta}$ , where all the pixel values are simultaneously scaled in magnitude by  $m$  and shifted in phase by  $\theta$ . That is, any measurement  $z$  is simply a representative of a whole class of possible equivalent measurements  $\{z s : s = m e^{j\theta} : m > 0, \forall \theta\}$ . Instead of being independent of each other, the real and imaginary components  $(x, y)$  of  $z$  covary in this equivalent class.

The covariance of the two components of complex-valued data has not been exploited in complex-valued deep learning. The common approach to learn a classifier invariant to scaling is to augment the training data with complex-valued scaling [19], [32], [52]. Such extrinsic data manipulation increases the amount of the training data and is rather ineffective: It takes a long time to train the model, yet the invariance is not guaranteed.

Our goal is to develop the invariance to complex-valued scaling as an intrinsic property of the neural network itself. We treat each complex-valued data sample as a point in a non-Euclidean space that respects the intrinsic geometry of complex numbers. We propose new convolution and fully connected (FC) layer functions that can achieve equivariance and invariance to complex-valued scaling.

There has been a long line of works that define convolution in a non-Euclidean space by treating each data sample as a function in that space [12], [17], [18], [21], [31], [53].

The challenge for defining such an equivariant convolution operator in the non-Euclidean space is the lack of a proper vector space structure. In the Euclidean space, we can move from one point to another using an element from the group of

Manuscript received August 26, 2019; revised January 29, 2020 and June 10, 2020; accepted September 29, 2020. This work was supported in part by Berkeley Deep Drive and in part by DARPA. (Corresponding author: Stella X. Yu.)

The authors are with the International Computer Science Institute (ICSI), University of California at Berkeley (UC Berkeley), Berkeley, CA 94720 USA (e-mail: stellayu@berkeley.edu).

Color versions of one or more of the figures in this article are available online at <https://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TNNLS.2020.3030565

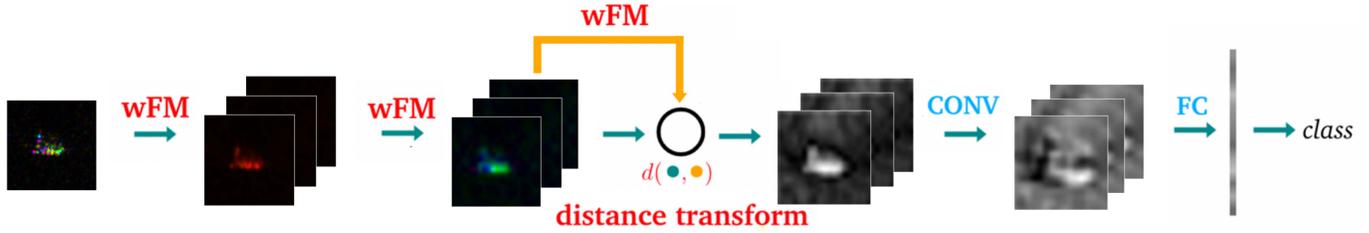


Fig. 1. Sample architecture of our SurReal model. First, the input is a complex-valued image; each pixel is color-coded with an HSV colormap according to its complex value, with the color intensity indicating the magnitude, the color hue indicating the phase, and the full constant saturation. Second, the first two layers are our proposed complex-valued convolution in terms of the weighted Fréchet mean (wFM) filtering on the manifold derived from the polar form of complex numbers. This convolution outputs complex-valued responses, and it is equivariant to complex-valued scaling. Each wFM layer could have multiple channels, each channel shown here as a complex-valued image. Pixelwise nonlinear activation functions, such as our proposed tReLU and  $G$ -transport, can be subsequently applied. Third, the third layer is a distance transformation layer, where the manifold distance between the feature map and its wFM is computed. This distance is invariant to complex-valued scaling. Fourth, once the representation becomes real-valued after the distance transformation layer, we could use any real-valued CNN layer functions for classification. Shown here is the real-valued convolution (CONV) layer followed by the FC layer toward the final softmax classification. With the built-in invariance to complex-valued scaling, our SurReal model can outperform real-valued baseline models on complex-valued data with a fraction of the baseline model size.

translations; the standard convolution is, thus, equivariant to the action of the group of translations. However, in the non-Euclidean space, e.g., a hypersphere, translation equivariance is no longer meaningful: Translation is not the group to move from one point to another on a hypersphere, but rotation is.

The concept of equivariance of an operator on a space is, thus, intimately related to the transitivity of a group of actions on that space. We say that a group  $G$  acts transitively on space if there exists a  $g \in G$  to go from one point to another on the space. The group of translations acts transitively on the Euclidean space, whereas the group of rotations acts transitively on a hypersphere. The group that transitively acts on the non-Euclidean space of the complex plane is nonzero scaling and planar rotations in the complex plane.

The manifold view of convolution as an operator with equivariance to transitive actions on that space applies to both the domain and the range of data, e.g., for an image, its pixel coordinates define the domain, and its pixel intensities define the range. Here, we focus on the range space of data in order to extend deep learning to complex-valued images and signals.

Our key insight is to represent a complex number in its polar form and define a Riemannian manifold on which complex scaling corresponds to the general transitive action group. When a data sample lies on a Riemannian manifold, there are previously established results for deep learning.

- 1) Convolution defined by the weighted Fréchet mean (wFM) [22] is equivariant to the group that naturally acts on that manifold [13].
- 2) Since wFM is nonlinear and acts like a contraction mapping [36] analogous to ReLU or sigmoid, nonlinear activation functions, such as ReLU, may not be needed.

We propose three types of complex-valued layer functions from the Riemannian geometric point of view.

- 1) *wFM*: A new convolution operator on the manifold for complex-valued data. It is equivariant to complex-valued scaling. The weights of wFM are to be learned.
- 2) *Tangent ReLU*: A new nonlinear activation function that applies ReLU to the projections in the tangent space of the complex-valued manifold. We also propose another option called  $G$ -transport that transports a point on

the complex manifold by an action in the scaling and rotation group.

- 3) *Distance Transform*: A new FC layer operator that computes the manifold distance between a feature map and its wFM. It is invariant to complex-valued scaling. The weights of wFM are to be learned.

The distance transform layer takes a complex-valued input to the real-valued domain, where any real-valued convolutional neural network (CNN) functions, such as standard convolutions and FC layers, can be subsequently used.

Fig. 1 shows a sample CNN architecture composed using our complex-valued layer functions. A complex-valued image first passes through two wFM complex-valued convolutional layers and then undergoes the distance transform. The resulting real-valued distances are subsequently fed into a real-valued CNN classifier with one convolution layer and one FC layer. Each convolutional layer is illustrated with a single channel response among a stack of many, with color images encoding complex-valued responses and grayscale images encoding real-valued responses.

Our complex-valued CNN has a group invariant property similar to the standard CNN on real-valued data. Existing methods extend the real-valued counterpart to the complex domain based on the form of functions, such as convolution or batch normalization [10], [49], [51], not on the property of functions, such as equivariance or linearity. Our complex-valued CNN is composed of layer functions with the desired equivariance and invariance properties that are essential for a real-valued CNN classifier in the Euclidean space; it is, thus, a theoretically justified analog of the real-valued CNN.

We compare our method with several baselines on two publicly available complex-valued data sets: MSTAR and RadioML. Our model consistently outperforms the real-valued CNN baseline, with fewer than 1% on MSTAR and 3% on RadioML of the baseline model parameters.

We, thus, name our approach SurReal (pun intended): a surprisingly lean complex-valued model that beats the real-valued CNN model. Our work has three major contributions.

- 1) We propose novel complex-valued layer functions with proven equivariance and invariance properties.
- 2) We extend our model to complex-valued residual CNNs.
- 3) We validate our method on classification experiments. Our SurReal CNNs outperform real- and complex-valued baselines at a fraction of their model sizes.

These results demonstrate significant benefits of proposing CNN layer functions in terms of desirable intrinsic properties on the complex plane as opposed to applying the standard CNN to the 2-D Euclidean embedding of complex numbers.

## II. RELATED WORKS

Complex numbers are powerful representations and concepts in mathematics, with intimate connections to geometry, topology, and differentiation [37]. They have a wide range of applications in physics and engineering.

Complex-valued data representations are widely used as a modeling choice to encode richer information than real-valued representations, especially for directional or cyclic data. Amin and Murase [1] learn a mapping from a finite range of real values to the unit circle in the complex plane. Cadieu and Olshausen [11] train a complex-valued sparse coding model to capture both edge structure and motion structure. Yu [54], [55] combines the confidence and size of a measurement in a single complex value and learns a global embedding from pairwise local measurements. Maire *et al.* [35] simultaneously encode both grouping and figure-ground ordering relationships between neighboring pixels and learn complex-valued pairwise pixel relationships from pixelwise figure-ground annotations. Reichert and Serre [43] use complex-valued neuronal units to model biologically plausible deep learning networks. Bruna and Mallat [7] and Bruna *et al.* [9] adopt wavelet transforms at earlier layers. Arjovsky *et al.* [2] adopt unitary weight matrices in hidden layers for better learning performance.

Traditional complex-valued data analysis utilizes higher order statistics, such as variance fractal dimension trajectory [30] and spectral analysis [44], to make adequate predictions.

Early neural network approaches have already noted that complex values have many nice mathematical properties that real-value data do not have, e.g., the complex identity theorem. Transformations from the input to the output can be more effectively learned with complex-valued networks instead of real-valued networks. Various complex-valued activation functions have been explored, although with little demonstration of their success in real data settings [4], [24], [29], [39].

Recent neural network approaches continue to build upon the theoretical advantages of complex-valued data to improve the convergence, stability, and generalization of neural networks [27], [38] and facilitate the noise-robust memory retrieval mechanisms in capsule networks [16]. Real-valued layer functions have also been extended to the complex domain according to the form of the functions, such as convolution, ReLU, and batch normalization [10], [49]–[51]. Complex-valued deep learning has also been extended to quaternion neural networks, as quaternions generalize the concept of complex values from 2-D to 3-D [42].

Recent graph convolution neural networks open up new computational models in the complex domain [8], [46]. Since

convolution in the spatial domain is equivalent to multiplication in the spectral domain, a natural extension of convolution to data defined on an arbitrary graph is to construct a convolutional filter in terms of multiplicative weights on the spectrum of the graph Laplacian [8]. The spectrum of the graph Laplacian is real-valued if the graph is undirected and complex-valued if it is directed [48].

Our SurReal complex-valued CNN is unique in utilizing the geometric property of the complex numbers and approaching complex-valued learning as a special task of deep learning on the Riemannian manifolds.

Existing methods, such as in [1], [35], [49], and [52], treat a complex value as a vector in the Euclidean space of  $\mathbf{R}^2$ . This choice, while straightforward, essentially destroys the covariant relationship between real and imaginary parts of a complex number. Naturally, complex-valued data, such as SAR, MRI, and RF, could be subject to complex-valued scaling without changing the underlying observation.

To deal specifically with complex numbers, we first separate and acknowledge the extrinsic scaling effect by asking the convolution operator to be equivariant to complex-valued scaling. For a CNN classifier, we design the distance transform layer to be invariant to complex-valued scaling.

Our SurReal CNN classifier can focus entirely on the discriminative information between classes, without the need to build up additional scale invariance by repeatedly training on data augmented with complex-valued scaling. Our SurReal model is, thus, a surprisingly lean complex-valued model that beats the real-valued CNN model on complex-valued data. An earlier preliminary version of this work was presented in [15].

## III. SCALING–ROTATION MANIFOLD FOR THE GEOMETRY OF COMPLEX NUMBERS

A crucial property of complex-valued data is complex-valued scaling ambiguity: The MRI or SAR images of the same scene could be related by the multiplication of a single complex number, depending on how the data are acquired. Complex-valued scaling can be captured by the scaling action on the magnitude and the rotation action on the phase.

Instead of treating the complex plane as the usual 2-D Euclidean space, we identify the nonzero complex plane as the product manifold of positive magnitudes and planar rotations. We show that scaling and rotation actions preserve the manifold distance defined on the nonzero complex manifold.

### A. Space of Complex Numbers

Let  $\mathbf{R}$  and  $\mathbf{C}$  denote the field of real numbers and complex numbers, respectively. We have

$$\mathbf{z} = x + iy \in \mathbf{C}, \quad \forall x, y \in \mathbf{R}. \quad (1)$$

According to this 2-D real-valued representation  $(x, y)$  of  $\mathbf{z}$ ,  $\mathbf{C}$  is a Riemannian manifold [5]. The distance induced by the canonical Riemannian metric is

$$d(\mathbf{z}_1, \mathbf{z}_2) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \quad (2)$$

the common Euclidean distance in the 2-D complex plane.

### B. Polar Form of Complex Numbers

Any nonzero complex number can be uniquely represented in the polar form, in terms of its magnitude and phase.

*Definition 1:*  $\forall \mathbf{z} \in \mathbf{C}$  and  $\mathbf{z} \neq \mathbf{0}$ , its polar form is

$$\mathbf{z} = |\mathbf{z}| \exp(i \angle \mathbf{z}) \quad (3)$$

$$\mathbf{z} \in \tilde{\mathbf{C}} = \mathbf{C} \setminus \{0 + i0\} \quad (4)$$

$$\text{magnitude: } |\mathbf{z}| = \sqrt{x^2 + y^2} \quad (5)$$

$$\text{phase: } \angle \mathbf{z} = \arctan(y, x) \quad (6)$$

where  $\exp$  is the exponential function and  $\arctan$  is the two-argument arc-tangent function that gives the angle in the complex plane between the positive  $x$  axis and the line from the origin to the point  $(x, y)$ .

### C. Scaling–Rotation Product Manifold for $\tilde{\mathbf{C}}$

Based on the polar form, we identify the nonzero complex plane  $\tilde{\mathbf{C}}$  as the product space of nonzero scaling and 2-D rotations

$$\tilde{\mathbf{C}} \iff \mathbf{R}^+ \times \text{SO}(2) \quad (7)$$

where  $\mathbf{R}^+$  is the manifold of positive reals and  $\text{SO}(2)$  is the manifold of planar rotations—a rotation Lie group.

We define a bijective mapping  $F$  that can go back and forth from the complex plane  $\tilde{\mathbf{C}}$  to the manifold space  $\mathbf{R}^+ \times \text{SO}(2)$

$$\mathbf{z} = |\mathbf{z}| \exp(i \angle \mathbf{z}) \xrightleftharpoons[F^{-1}]{} (|\mathbf{z}|, R(\angle \mathbf{z})) \quad (8)$$

$$R(\angle \mathbf{z}) = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix}. \quad (9)$$

Both spaces are parameterized by magnitude and phase; the phase is turned into a complex number with  $\exp(\cdot)$  for  $\tilde{\mathbf{C}}$  and into a 2-D rotation matrix with  $R(\cdot)$  for  $\mathbf{R}^+ \times \text{SO}(2)$ .

### D. Manifold Distance for $\tilde{\mathbf{C}}$

The exponential and logarithmic maps are respectively  $\exp$  and  $\log$  for  $\mathbf{R}^+$ , matrix exponential  $\text{expm}$ , and matrix logarithm  $\text{logm}$  for  $\text{SO}(2)$ .

*Definition 2:* The matrix exponential and logarithm of matrix  $X$  are defined, respectively, as

$$\text{expm}(X) = \sum_{n=0}^{\infty} \frac{X^n}{n!}$$

$$X = \text{logm}(Y) \text{ if and only if } Y = \text{expm}(X).$$

The distance on this product manifold in (2) becomes

$$d(\mathbf{z}_1, \mathbf{z}_2) = \sqrt{\log^2 \frac{|\mathbf{z}_2|}{|\mathbf{z}_1|} + \|\text{logm}(R(\angle \mathbf{z}_2) R(\angle \mathbf{z}_1)^{-1})\|^2}. \quad (10)$$

### E. Scaling–Rotation Is Transitive on $\tilde{\mathbf{C}}$

The complex plane  $\tilde{\mathbf{C}}$  as identified with  $\mathbf{R}^+ \times \text{SO}(2)$  is a Riemannian homogeneous space [26]. We define transitive actions that move a point around on the manifold [20].

*Definition 3:* Given a (Riemannian) manifold  $\mathcal{M}$  and a group  $G$  with identity element  $e$ , we say that  $G$  acts on  $\mathcal{M}$  (from the left) if there exists a mapping  $L : G \times \mathcal{M} \rightarrow \mathcal{M}$  given by  $(g, X) \mapsto g.X$  that satisfies two conditions.

1) *Identity:*  $L(e, X) = e.X = X$ .

2) *Compatibility:*  $(gh).X = g.(h.X) \forall g, h \in G$ .

An action is called transitive if and only if, given  $X, Y \in \mathcal{M}$ , there exists an element  $g \in G$  such that  $Y = g.X$ .

It is straightforward to verify that scaling and rotation in  $\mathbf{R}^+ \times \text{SO}(2)$  satisfy the identity and compatibility conditions on  $\tilde{\mathbf{C}}$ . It is also a transitive group action: for any complex numbers  $\mathbf{z}_1, \mathbf{z}_2 \in \tilde{\mathbf{C}}$ , there always exists a relative scaling (of the magnitude) and rotation (of the phase) that maps  $\mathbf{z}_1$  to  $\mathbf{z}_2$ .

*Proposition 1:* The scaling–rotation Lie group  $\mathbf{R}^+ \times \text{SO}(2)$  transitively acts on  $\tilde{\mathbf{C}}$ , and the action  $g$  to take  $\mathbf{z}_1$  to  $\mathbf{z}_2$  is

$$g = \left( \frac{|\mathbf{z}_2|}{|\mathbf{z}_1|}, R(\angle \mathbf{z}_2) R(\angle \mathbf{z}_1)^{-1} \right) \in \mathbf{R}^+ \times \text{SO}(2). \quad (11)$$

### F. Scaling–Rotation Is Isometric on $\tilde{\mathbf{C}}$

We now show that scaling and rotation actions preserve our manifold distance.

*Proposition 2:* The scaling and rotation Lie group is isometric on the complex plane  $\tilde{\mathbf{C}}$ :  $\forall \mathbf{z}_1, \mathbf{z}_2 \in \tilde{\mathbf{C}}, g \in \mathbf{R}^+ \times \text{SO}(2)$

$$d(g.\mathbf{z}_1, g.\mathbf{z}_2) = d(\mathbf{z}_1, \mathbf{z}_2) \quad (12)$$

where  $d$  is the manifold distance defined in (10).

*Proof:* We use the definition of  $d$  and the property that the 2-D rotation group  $\text{SO}(2)$  is Abelian:  $\forall A, B \in \text{SO}(2)$ ,  $AB = BA$ . Let  $g = (r, A) \in \mathbf{R}^+ \times \text{SO}(2)$ . We have

$$\begin{aligned} d(g.\mathbf{z}_1, g.\mathbf{z}_2) &= \sqrt{\log^2 \frac{r|\mathbf{z}_2|}{r|\mathbf{z}_1|} + \|\text{logm}(AR(\angle \mathbf{z}_2)(AR(\angle \mathbf{z}_1))^{-1})\|^2} \\ &= \sqrt{\log^2 \frac{|\mathbf{z}_2|}{|\mathbf{z}_1|} + \|\text{logm}(R(\angle \mathbf{z}_2)A(R(\angle \mathbf{z}_1)A)^{-1})\|^2} \\ &= \sqrt{\log^2 \frac{|\mathbf{z}_2|}{|\mathbf{z}_1|} + \|\text{logm}(R(\angle \mathbf{z}_2)R(\angle \mathbf{z}_1)^{-1})\|^2} = d(\mathbf{z}_1, \mathbf{z}_2) \end{aligned}$$

completing the proof.  $\square$

## IV. CNN LAYER FUNCTIONS ON COMPLEX MANIFOLD

We propose complex-valued CNN layer functions based on the scaling–rotation manifold view of complex numbers. Each layer function transforms the representation with a certain property (e.g., equivariance or invariance) on the manifold.

We define first a convolution operator on the manifold that is equivariant to complex-valued scaling. We then define nonlinear activation functions and FC layer functions that are invariant to complex-valued scaling. A CNN composed of such layer functions would then be intrinsically invariant to complex-valued scaling.

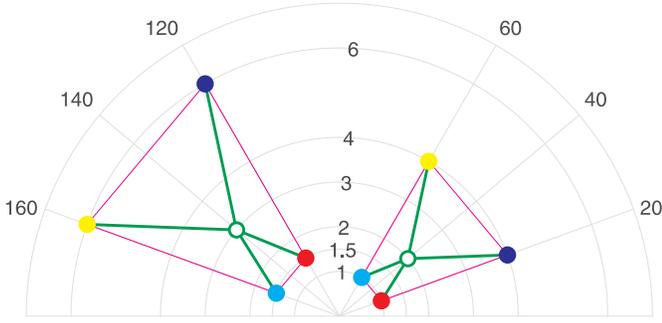


Fig. 2. Our complex-valued convolution in terms of wFM on the  $\mathbf{R}^+ \times \text{SO}(2)$  manifold is equivariant to complex-valued scaling. Consider four numbers in the complex plane, marked by four colored points on a small magenta trapezoid. Their equally weighted wFM (marked by the green circle) sits inside the trapezoid at the geometric mean of their magnitudes and the mean of their phases. When the four complex numbers are multiplied by  $1.5 \exp(i(100\pi/180))$ , the points are scaled by 1.5 and rotated by  $100^\circ$ , moving to the larger trapezoid. The new wFM is simply the old wFM transported by the same movement.

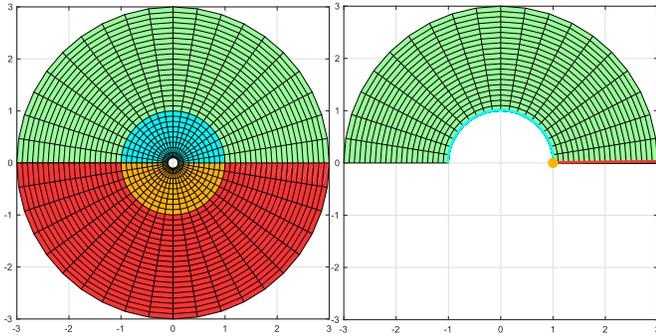


Fig. 3. Our tReLU extends ReLU from the real line to the complex plane  $\tilde{\mathbf{C}}$ , by applying ReLU in the tangent space of  $\mathbf{R}^+ \times \text{SO}(2)$ . It rectifies the magnitude by 1 and phase by 0, creating four corresponding regions before (left) and after (right) the mapping. Green region: points retain both their magnitudes and phases. Cyan region: points retain their phases with their magnitudes rectified to 1. Red region: points retain their magnitudes with their phases rectified to 0. Brown region: points are rectified in both the magnitude and the phase, all to the same point  $1 + i \cdot 0$ .

### A. Complex-Valued Convolutional Layer Function

The standard convolution, denoted by  $*$  and  $n$  weights  $\{w_k\}$  over  $n$  neighboring points  $\{x_k\}$ , is simply the weighted average of real numbers in the Euclidean space

$$\{w_k\} * \{x_k\} = \sum_{k=1}^n w_k x_k. \quad (13)$$

We extend this concept to points on a manifold.

1) *Fréchet Mean on the Manifold*: The weighted average of  $n$  points on a Riemannian manifold is called the wFM [22]. We define the complex convolution, denoted by  $\tilde{*}$ , as wFM on the scaling–rotation manifold for complex values (center circles in Fig. 2)

$$\{w_k\} \tilde{*} \{z_k\} = \text{wFM}(\{w_k\}, \{z_k\}) \quad (14)$$

$$\text{wFM}(\{w_k\}, \{z_k\}) := \arg \min_{\mathbf{m} \in \tilde{\mathbf{C}}} \sum_{k=1}^n w_k d^2(\mathbf{z}_k, \mathbf{m}) \quad (15)$$

$$\sum_{k=1}^n w_k = 1, \quad w_k \geq 0 \quad (16)$$

where  $d$  is the manifold distance in (10).

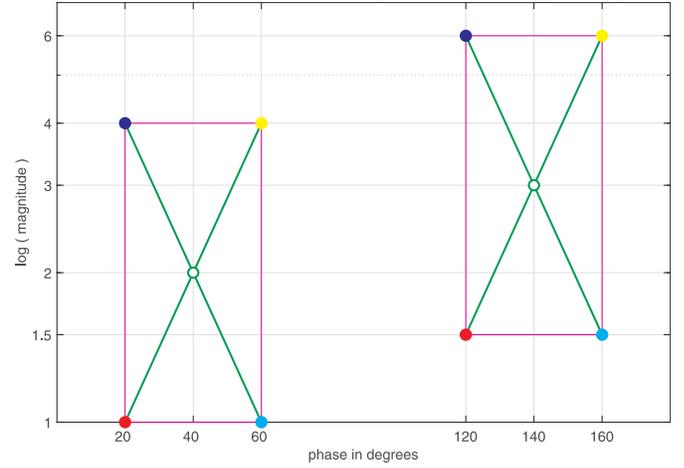


Fig. 4. Our distance transform is invariant to complex-valued scaling. We plot Fig. 2 in the tangent space of  $\mathbf{R}^+ \times \text{SO}(2)$ , with magnitude on a log scale for the y-axis and phase for the x-axis. The distance  $d$  on the manifold is directly measured by the Euclidean distance in this space. When the four complex numbers are multiplied by  $1.5 \exp(i(100\pi/180))$ , the points and their wFM are simply translated. The distances between the points and their wFM, thus, remain the same. Note that while this direct phase  $\theta$  representation is intuitive, our rotation matrix  $R(\theta)$  can more easily handle phase representation discontinuity at, e.g.,  $\pm\pi$ , where  $\theta(-\pi, +\pi]$  for a unique determination of  $\theta$ .

We contrast our complex-valued wFM convolution  $\tilde{*}$  with the standard real-valued convolution  $*$ .

- While the output of  $\tilde{*}$  is complex-valued, the weights  $\{w_k\}$  are real-valued, just like the weights for  $*$ .
- While the weights of  $*$  can be arbitrary, the weights of  $\tilde{*}$  are all nonnegative and summed up to 1. This convexity constraint ensures that the wFM of  $n$  points on a manifold stays on the manifold.
- While the output of  $*$  is simply the weighted average, the output of  $\tilde{*}$  is the minimizer to a weighted least-squares problem, i.e., the data mean that minimizes the weighted variance. There is no closed-form solution to wFM; however, there is a provably convergent  $n$ -step iterative solution [45].
- If  $d$  is the manifold distance in (2) for the Euclidean space that is also Riemannian, then wFM has exactly the weighted average as its closed-form solution. That is, our wFM convolution on the Euclidean manifold is reduced to the standard convolution, although with the additional convexity constraint on the weights.

2) *Convolutional wFM Layer*: As for the standard convolution, our weights  $\{w_k\}$  for wFM are parameters learnable through stochastic gradient descent (SGD), with the additional convexity constraint on  $\{w_k\}$ .

Each set of weights  $\{w_k\}$  defines a single wFM channel in our convolutional layer, and each layer has multiple channels. We follow the CNN convolution convention for images, where the convolutional kernel spans a local spatial neighborhood but all the channels. If there are 10 input channels of  $50 \times 50$  pixels, to produce 20 output channels with  $5 \times 5$  spatial kernels, we need to learn 20 sets of  $5 \times 5 \times 10$  weights.

3) *Equivariance of wFM to  $\mathbf{R}^+ \times \text{SO}(2)$* : We have shown that the scaling–rotation Lie group transitively acts on  $\tilde{\mathbf{C}}$  and

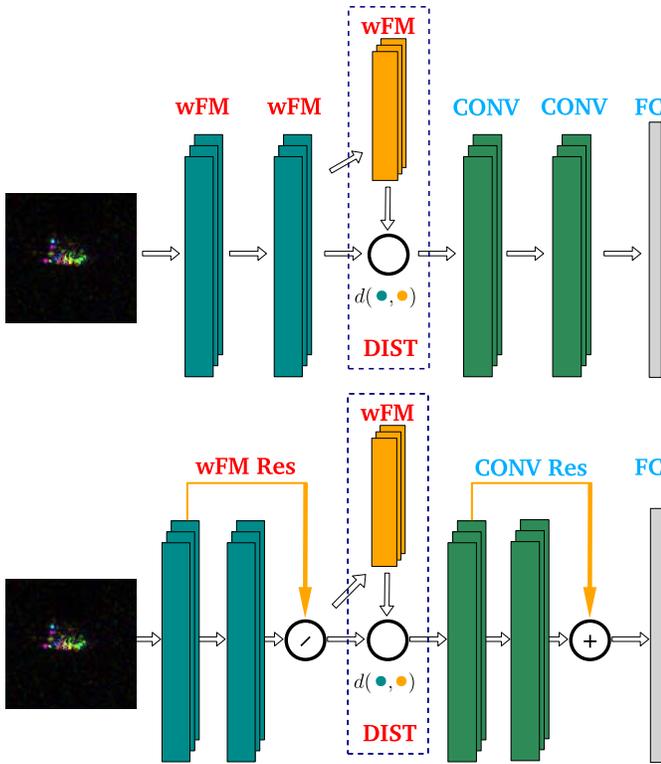


Fig. 5. Our complex-valued CNN with residual connections allows manifold valued representations to be combined across different depths. Top: sample CNN with complex-valued convolution (wFM), distance transform (DIST), real-valued convolution (CONV), and FC layers. Bottom: to create a residual structure (Res), we add skip connections between two adjacent convolutional layers. We first spatially align their feature maps via convolution (wFM for complex-valued and CONV for real-valued) and then combine them via channelwise concatenation for complex-valued data and addition for the real-valued data. The only difference between real- and complex-valued residual blocks is how the feature maps are combined: Addition is a vector space operator that does not apply to two points on a non-Euclidean manifold.

is isometric (see Proposition 2). We use this result to prove that our wFM is equivariant to complex-valued scaling.

*Proposition 3:* The complex-valued convolution  $\tilde{*}$  in (14) is equivariant to the action of  $\mathbf{R}^+ \times \text{SO}(2)$ :  $\forall g \in \mathbf{R}^+ \times \text{SO}(2)$

$$\{w_k\} \tilde{*} \{g \cdot \mathbf{z}_k\} = g \cdot (\{w_k\} \tilde{*} \{\mathbf{z}_k\}). \quad (17)$$

*Proof:* Let  $g \in \mathbf{R}^+ \times \text{SO}(2)$  and  $\mathbf{o} = \{w_k\} \tilde{*} \{\mathbf{z}_k\}$ .  $\forall \mathbf{m} \in \tilde{\mathcal{C}}$

$$\sum_{k=1}^n w_k d^2(g \cdot \mathbf{z}_k, g \cdot \mathbf{o}) = \sum_{k=1}^n w_k d^2(\mathbf{z}_k, \mathbf{o}) \leq \sum_{k=1}^n w_k d^2(\mathbf{z}_k, \mathbf{m})$$

since  $g$  preserves the distance and  $\mathbf{o}$  is the minimizer over  $\{\mathbf{z}_k\}$ . Therefore,  $g \cdot \mathbf{o}$  is the minimizer over  $\{g \cdot \mathbf{z}_k\}$ .  $\square$

Fig. 2 illustrates how wFM is equivariant to rotation and scaling. For each trapezoid, the center circle marks the wFM of the four corner points. If the trapezoid is transported using a particular scaling and rotation action, then the center wFM is also transported by the same action.

## B. Nonlinear Activation Functions

The wFM convolution is a contractive mapping, an effect of a nonlinear activation function. Nevertheless, for stronger

nonlinearity and acceleration in optimization during learning, we propose two activation functions from the manifold perspective: tangent ReLU (tReLU) and  $G$ -transport.

1) *Tangent ReLU:* The tangent space of a manifold is a vector space that contains the possible directions for tangentially passing through a point on the manifold. It could be regarded as a local Euclidean approximation of the manifold. A pair of logarithmic and exponential maps establish the correspondence between the manifold and the tangent space.

We extend ReLU to the complex plane  $\tilde{\mathcal{C}}$  by applying ReLU in the tangent space of  $\mathbf{R}^+ \times \text{SO}(2)$  manifold. Our tReLU is composed of three steps.

- 1) Apply logarithmic maps to go from a point in  $\tilde{\mathcal{C}}$  to a point in its tangent space. The mapping is  $\log$  for  $r \in \mathbf{R}^+$  and  $\log m$  for  $R(\theta) \in \text{SO}(2)$ , which produces a skew symmetric matrix. We choose the principal log map for  $\text{SO}(2)$  in the range of  $\theta \in (-\pi, \pi]$

$$\log m(R(\theta)) = \theta \cdot \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}. \quad (18)$$

- 2) Apply ReLU in the tangent space of  $\tilde{\mathcal{C}}$ . ReLU is well defined for a real-valued scalar, such as  $\log(r)$  for  $\mathbf{R}^+$

$$\text{ReLU}(x) = \max(x, 0), \quad \text{e.g., } x = \log(r). \quad (19)$$

We extend ReLU to  $\log m(R(\theta))$  for  $\text{SO}(2)$  since it is just the real-valued  $\theta$  scaled by a constant matrix

$$\text{ReLU}(\log m(R(\theta))) = \max(\theta, 0) \cdot \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}. \quad (20)$$

- 3) Apply exponential maps to come back to  $\tilde{\mathcal{C}}$  from the tangent space. We can simplify the three-step tReLU as

$$r \xrightarrow{\text{tReLU}} \exp(\text{ReLU}(\log(r))) = \max(r, 1) \quad (21)$$

$$R(\theta) \xrightarrow{\text{tReLU}} \exp m(\text{ReLU}(\log m(R(\theta)))) = R(\max(\theta, 0)). \quad (22)$$

Fig. 3 shows that our manifold perspective of  $\tilde{\mathcal{C}}$  leads to a nontrivial extension of ReLU from the real line to the complex plane, partitioning  $\tilde{\mathcal{C}}$  into four regions, separated by  $r = 1$  and  $\theta = 0$ . Those with magnitudes smaller than 1 are rectified to 1, and those with negative phases are rectified to 0.

2) *G-Transport:* A nonlinear activation function is a general mapping that transforms the range of responses. We consider a general alternative that simply transports all the values in a feature channel via an action in the group  $G = \mathbf{R}^+ \times \text{SO}(2)$ . We only need to learn one global scaling and rotation per feature channel, which corresponds to learning one complex-valued multiplier per channel at a certain depth layer.

## C. Fully Connected Layer Functions

For classification tasks, having equivariance of convolution and range compression of nonlinear activation functions is not enough; we need the final representation of a CNN to be invariant to variations within each class.

In a standard CNN classifier, the entire network is invariant to the action of the group of translations, achieved by the FC

layer. Likewise, we develop an FC layer function on  $\tilde{\mathbf{C}}$  that is invariant to the action of  $\mathbf{R}^+ \times \text{SO}(2)$ .

Since the manifold distance  $d$  is invariant to actions in  $G = \mathbf{R}^+ \times \text{SO}(2)$ , we propose the distance between each point of a set and their wFM, which is equivariant to  $G$ , as a new FC function on  $\tilde{\mathbf{C}}$ .

1) *Distance Transform FC Layer*: Let the input be  $s$  pixels of  $c$  channels each. We perform a global integration over all these  $s \cdot c$  complex values  $\{\mathbf{t}_k\}$ . Given  $s \cdot c$  weights  $\{w_k\}$  for these individual numbers, we first calculate their wFM  $\mathbf{m}$  and then compute the distance  $u_k$  from  $\mathbf{t}_k$  to the mean  $\mathbf{m}$

$$\mathbf{m} = \{w_k\} \tilde{*} \{\mathbf{t}_k\} \quad (23)$$

$$u_k = d(\mathbf{t}_k, \mathbf{m}). \quad (24)$$

The output is real-valued and of the same size as the input. The weights  $\{w_k\}$  are the parameters to be learned, and there could be multiple sets of such weights at this layer.

*Proposition 4*: The distance to the wFM, defined in (23) and (24), is invariant to the action of  $G = \mathbf{R}^+ \times \text{SO}(2)$ .

*Proof*: Per Propositions 2 and 3,  $\forall g \in G$ , we have

$$\begin{aligned} d(g \cdot \mathbf{t}_k, \{w_k\} \tilde{*} \{g \cdot \mathbf{t}_k\}) \\ &= d(g \cdot \mathbf{t}_k, g \cdot (\{w_i\} \tilde{*} \{\mathbf{t}_k\})) \quad \text{equivariance of wFM} \\ &= d(\mathbf{t}_k, \{w_k\} \tilde{*} \{\mathbf{t}_k\}) \quad \text{invariance of distance} \end{aligned}$$

completing the proof.  $\square$

Fig. 4 replots Fig. 2 in the  $(\log(r), \theta)$  space, which corresponds to the tangent space of  $\mathbf{R}^+ \times \text{SO}(2)$  where the manifold distance can be directly visualized as the Euclidean distance. When the four corners of the trapezoid are scaled and rotated, the trapezoid is simply translated along  $(\log(r), \theta)$  axes. The distance from points to their wFM remains the same.

Since the output of the distance transform layer is real-valued, we can subsequently use any existing layer functions of a real-valued CNN classifier. Fig. 1 shows a sample architecture of our complex-valued CNN, where two successive wFM convolutional layers are followed by a distance transform FC layer, a standard convolutional layer, and an FC layer for final softmax classification.

#### D. Complex-Valued Residual Layer Function

A standard CNN with residual layers, such as ResNet [25], outperforms the one without. Residual layers are useful for preventing exploding/vanishing gradients in deep networks, by utilizing skip connections to jump over some layers. The skip connections between layers add the outputs from previous layers to the outputs of stacked layers.

While addition is natural for combining layers in the field of real numbers, it does not make sense in the field of complex numbers: We can add two vectors in the Euclidean space, but we cannot add two points on a non-Euclidean manifold.

Here, we propose a complex-valued residual layer function by retaining the skip connection concept without the addition to combine the outputs from different layers. Let a feature layer  $\mathbf{f}(s, c)$  be specified by the number of pixels  $s$  and the number of channels  $c$ . Consider two feature layers  $\mathbf{f}_1(s_1, c_1)$  and  $\mathbf{f}_2(s_2, c_2)$ ,  $s_1 < s_2$ , with one layer through skip connections. In order to combine them, we first use the wFM

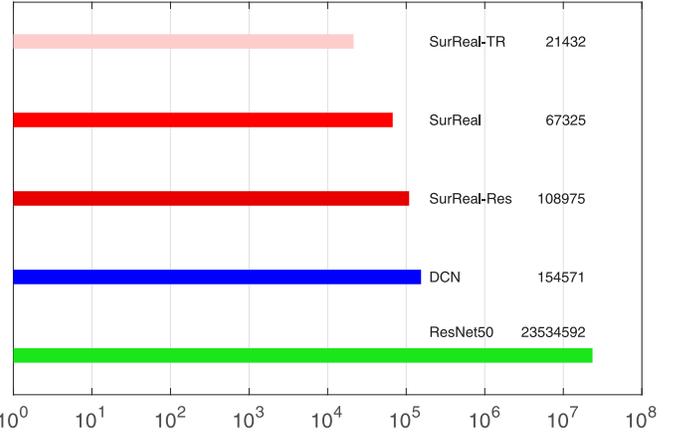


Fig. 6. Our SurReal models for MSTAR are much leaner than baselines. Each model size is plotted as a horizontal bar on a log scale, labeled with the model name and the number of parameters on the right. ResNet50 is the real-valued baseline and the largest with 23.5M parameters. DCN is the complex-valued baseline, with 154k parameters at 0.7% of ResNet50. Our SurReal residual network has 109k parameters at 0.5% (71%) of ResNet50 (DCN). Our basic SurReal CNN has 67k parameters at 0.3% (44%) of ResNet50 (DCN). With the tensor ring implementation for convolutions, our SurReal CNN could be further reduced to 21k parameters at 0.1% (14%) of ResNet50 (DCN).

convolution to bring the spatial dimension of  $\mathbf{f}_2$  from  $s_2$  to  $s_1$  and then concatenate the two sets of spatially aligned features

$$\text{align spatially: } \mathbf{f}_2(s_2, c_2) \xrightarrow{\tilde{*}} \tilde{\mathbf{f}}_2(s_1, c_2) \quad (25)$$

$$\text{concatenate: } [\mathbf{f}_1(s_1, c_1) \mid \tilde{\mathbf{f}}_2(s_1, c_2)] \rightarrow \mathbf{f}(s_1, c_1 + c_2). \quad (26)$$

Once combined, we can treat them as the input and apply any wFM convolution as desired.

Fig. 5 shows that we can simply replace two complex-valued convolution layers with such a residual block connecting and combining their outputs and build a residual complex-valued convolution network. The only difference with the real-valued residual block is that the combination is channelwise concatenation for a non-Euclidean manifold instead of an addition for a vector space.

We can optionally further reduce the number of parameters for convolution using the tensor ring decomposition [56]. A  $c$ -dimensional convolutional filter  $W$  of size  $n_1 \times \dots \times n_c$  can be decomposed into  $c$  smaller rank  $b$  tensors, each of the form  $T_k$  with size  $b \times n_k \times b$  such that  $\forall k_1, \dots, k_c$

$$W(k_1, \dots, k_c) = \text{trace}(T_1(:, k_1, :) \times \dots \times T_c(:, k_c, :)) \quad (27)$$

where  $\times$  denotes the matrix multiplication. Such a tensor factorization needs  $b^2 \sum_{k=1}^c n_k$  parameters for all the tensors  $\{T_k\}$  instead of  $\prod_{k=1}^c n_k$  parameters for the original  $W$ . Tensor ring decomposition can achieve arbitrary approximation precision [40, p. 2299, Th. 2.2].

## V. EXPERIMENTS

We compare our SurReal complex-valued classifier against two baselines: 1) the first baseline is a real-valued CNN classifier, such as ResNet50, which ignores the geometry of complex numbers and treats each complex value as two independent real numbers and 2) the second baseline is the deep complex networks (DCNs), which extends real-valued

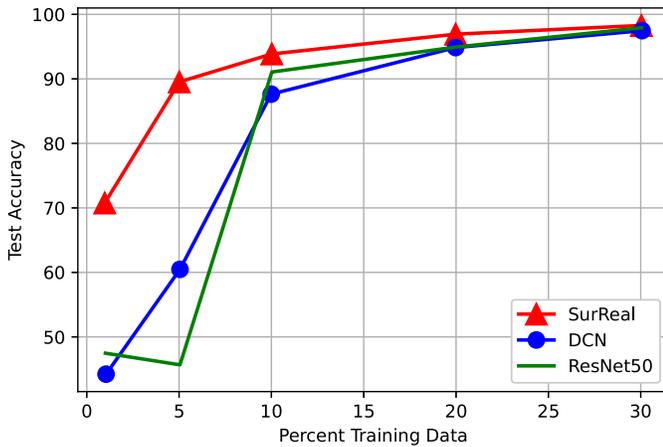


Fig. 7. Our SurReal CNN significantly outperforms real- and complex-valued baselines on MSTAR target recognition when less data are used for training. We split the data into training and test sets at varying proportions and plot the test accuracy. When there is enough training data, e.g., at 30% training and 70% testing, all three models perform similarly with 98% accuracy. When there is less training data, e.g., at 5% training and 95% testing, SurReal is much better with a test accuracy of 90% over DCN’s 60% and ResNet50’s 45%, demonstrating the effectiveness of our complex-valued model.

TABLE I  
MSTAR DATA SET SIZE: NUMBER OF IMAGES PER CLASS

C0	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10
1285	429	6694	451	1164	1415	573	572	573	1401	1159

CNN layer functions to the complex domain by the form of the functions, such as complex-valued convolution, batch-normalization, nonlinear activation, and weight initialization [49]. The specific DCN models used in [49] are very large, on the order of one million parameters. To help directly compare complex-valued layer functions, we adopt the same SurReal Residual architecture for the DCN baseline but replacing all the convolution layers (complex-valued wFM and real-valued convolution) and nonlinear activation functions with DCN’s proposed counterparts.

We experiment on two complex-valued data sets: SAR image data set MSTAR [28] and synthetic RF signal data set RadioML [41]. All the models are trained on a GeForce RTX 2080 GPU for 120 epochs, using Adam optimizer and cross-entropy loss.

### A. MSTAR Classification

1) *MSTAR Data Set*: There are a total of 15 716 complex-valued X-band SAR images, distributed unevenly over 11 classes: The first ten classes contain different target vehicles, and the last one class contains background clutter (see Table I for the total number of images per class). We take the  $100 \times 100$  center crop of each image and convert the complex value of each pixel into the polar form.

2) *Real-Valued CNN Baseline*: ResNet50 [25] is widely successful on real-valued image classification, and it is also used as a baseline in [47].

TABLE II  
SURREAL CNN IN DETAILED LAYER SPECIFICATION

Layer Type	Input Shape	Kernel	Stride	Output Shape
wFM CONV	[2, 1, 100, 100]	$5 \times 5$	2	[2, 20, 48, 48]
G-transport	[2, 20, 48, 48]	-	-	[2, 20, 48, 48]
wFM CONV	[2, 20, 48, 48]	$5 \times 5$	2	[2, 20, 22, 22]
G-transport	[2, 20, 22, 22]	-	-	[2, 20, 22, 22]
DIST	[2, 20, 22, 22]	-	-	[20, 22, 22]
CONV	[20, 22, 22]	$5 \times 5$	1	[30, 18, 18]
BN+ReLU	[30, 18, 18]	-	-	[30, 18, 18]
MaxPool	[30, 18, 18]	$2 \times 2$	2	[30, 9, 9]
CONV	[30, 18, 18]	$5 \times 5$	3	[30, 2, 2]
BN+ReLU	[30, 2, 2]	-	-	[30, 2, 2]
CONV	[30, 2, 2]	$2 \times 2$	1	[30, 1, 1]
BN+ReLU	[30, 1, 1]	-	-	[30, 1, 1]
FC	[30]	-	-	[50]
FC	[50]	-	-	[11]

The network has two layers of complex-valued convolution (wFM) and nonlinear activation (G-transport), distance transform (DIST), and then a real-valued CNN classifier composed with standard convolution (CONV), batch normalization (BN) and ReLU, max pooling (MaxPool), and fully connected (FC) layers. An input or output shape of 4 dimensions indicates complex-valued data, with each complex number represented by two values: magnitude and phase. For example, [2, 1, 100, 100] means a complex-valued 1-channel  $100 \times 100$  image. A shape of 3 dimensions indicates real-valued data. In our network, DIST is the depth layer that turns a complex-valued representation into a real-valued one; it separates the complex-valued and real-valued representations in the SurReal CNN.

3) *Two SurReal CNN Architectures*: Table II lists detailed layer specification of our basic model. Table III adds residual connections. Both models have two complex-valued convolutions (with nonlinear activation), one distance transform layer, two real-valued convolutions (with batch normalization and ReLU), max pooling, and two FC layers. While we have listed G-transport in the two tables, we have also tried tReLU as the complex-valued nonlinear activation function; their difference is insignificant in initial experiments on the MSTAR data set, and we focus on G-transport for its simplicity.

4) *Model Size Comparison*: While ResNet50 and DCN have 23 million and 155k parameters, respectively, our SurReal CNN has 67k parameters, and SurReal residual CNN has 109k parameters. We can further reduce the parameter count by implementing convolutions with tensor ring decomposition [40], [56]. Fig. 6 plots these model sizes on a log scale. The saving is substantial: our SurReal CNN is less than 0.1% of the real-valued baseline and 44% of the complex-valued baseline.

5) *Task 1: Ten-Class Target Recognition*: For the ten target classes, we split all the data into five varying proportions of 1%, 5%, 10%, 20%, and 30% for training and the rest for testing. Fig. 7 shows that our SurReal significantly outperforms DCN and ResNet50, especially when a small percentage of training data is used. At 5% training and 95% testing, the accuracy is 90% for SurReal, 60% for DCN, and 45% for ResNet50.

TABLE III  
SURREAL RESIDUAL CNN IN DETAILED LAYER SPECIFICATION

Layer Type	Input Shape 1	Input Shape 2	Kernel	Stride	Output Shape
wFM CONV	[2, 1, 100, 100]	-	5 × 5	2	[2, 20, 48, 48]
G-transport	[2, 20, 48, 48]	-	-	-	[2, 20, 48, 48]
wFM CONV	[2, 20, 48, 48]	-	5 × 5	2	[2, 20, 22, 22]
wFM Res	[2, 20, 48, 48]	[2, 20, 22, 22]	5 × 5	2	[2, 20, 22, 22]
G-transport	[2, 20, 22, 22]	-	-	-	[2, 20, 22, 22]
DIST	[2, 20, 22, 22]	-	-	-	[20, 22, 22]
CONV	[20, 22, 22]	-	5 × 5	1	[30, 18, 18]
BN+ReLU	[30, 18, 18]	-	-	-	[30, 18, 18]
CONV Stack	[30, 18, 18]	-	1, 3, 1	1	[30, 18, 18]
CONV Res	[30, 18, 18]	[30, 18, 18]	-	-	[40, 18, 18]
Maxpool	[40, 18, 18]	-	2 × 2	2	[40, 9, 9]
CONV	[40, 18, 18]	-	5 × 5	3	[50, 2, 2]
BN+ReLU	[50, 2, 2]	-	-	-	[50, 2, 2]
CONV Stack	[50, 2, 2]	-	1, 3, 1	1	[50, 2, 2]
CONV Res	[50, 2, 2]	[50, 2, 2]	-	-	[60, 2, 2]
CONV	[60, 2, 2]	-	2 × 2	1	[70, 1, 1]
BN+ReLU	[70, 1, 1]	-	-	-	[70, 1, 1]
FC	[70]	-	-	-	[30]
FC	[30]	-	-	-	[11]

Our SurReal residual CNN utilizes both complex-valued residual blocks (Fig.5) as well as real-valued residual blocks. In CONV Stack, we stack three convolutions with  $1 \times 1$ ,  $3 \times 3$ ,  $1 \times 1$  kernels respectively. We zero pad the inputs on the  $3 \times 3$  convolution to preserve the spatial dimensions.

TABLE IV  
MODEL ACCURACY COMPARISON ON MSTAR-L AND MSTAR-S

Test Accuracy (%)	ResNet50	DCN	SurReal	SurReal-Res
MSTAR-L	99.1	98.9	99.1	<b>99.2</b>
MSTAR-S	97.4	93.3	97.7	<b>98.4</b>
Difference S-L	-1.7	-6.6	-1.4	<b>-0.8</b>

Our SurReal CNN and its residual version consistently outperform the real-valued baseline ResNet50 and complex-valued baseline DCN. SurReal-Res retains the accuracy the most when the overall data size is reduced, demonstrating better generalization capability from small data.

6) *Task 2: 11-Class Classification*: We also include the remaining clutter class that contains miscellaneous background images. We create two random subsets, large (L) and small (S), and the small set of 6295 images is contained entirely in the large set of 12610 images. Fig. 8 shows the number of instances across 11 classes and in training/testing splits.

Table IV shows that all the models perform at high accuracy of 99% for the large data set. The performance drops as the overall data size is reduced by half in the small data set, but the drop is the least at  $-0.8\%$  for SurReal-Res, followed by  $-1.4\%$  for SurReal,  $-1.7\%$  for ResNet50, and the most at  $-6.6\%$  for DCN. Our results against the two baselines suggest that it is both the residual connections and more

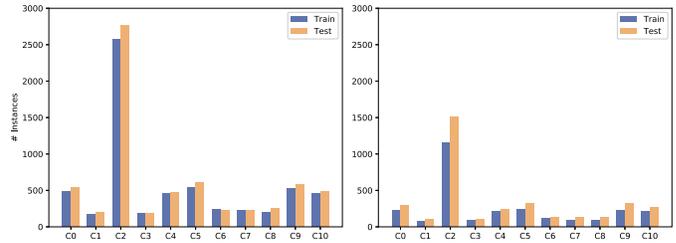


Fig. 8. MSTAR large (L) and small (S) subsets have highly imbalanced classes. C2 is the largest class, C0, C4, C5, C9, and C10 come next at about 20% of the size of C2, and C1, C3, C6, C7, and C8 are at about 8% of the size of C2.

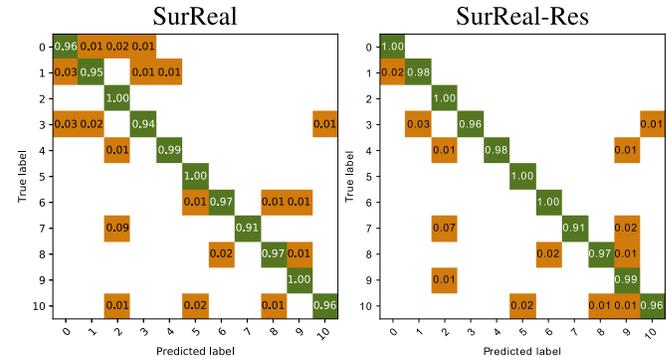


Fig. 9. Confusion matrix for our SurReal CNN (left) and SurReal-Res CNN (right) on MSTAR-S. First, the data set is small and highly imbalanced across classes (see Fig. 8 Right). The model achieves better accuracy for classes with more instances: The accuracy is 100% for C2 and 94% for C1 and C3. Overall, the accuracy gap is small, considering that the size of C1 and C3 is only 8% of the size of C2. Second, the residual connections help further clear up the confusion between classes. The matrix becomes more strongly diagonal.

importantly how we handle complex-valued data that deliver more generalizing performance from smaller training data.

Fig. 9 shows the confusion matrix between classes on MSTAR-S. In general, the more the training instances in the class, the least the confusion with other classes at the test time. However, despite the significant class imbalance, the performance gap is small between the minority and majority classes. Residual connections help clear up more confusion.

Fig. 10 shows the sample channel responses from our SurReal CNN on MSTAR-S images. With two complex-valued wFM convolutions, followed by distance transform and real-valued convolution, the representation for each input SAR image quickly becomes more distinctive across classes, facilitating accurate discrimination.

## B. RadioML Classification

1) *RadioML Data Set*: They are synthetically generated radio signals with modulation operating over both voice and text data. Noise is added further for channel effects. Each signal has 128 time samples and is tagged with a signal-to-noise ratio (SNR), in the range of  $[-20, 18]$  with an increment step of 2. There are 11 modulation modes, of which BPSK, QPSK, 8PSK, 16QAM, 64QAM, BFSK, CPFSK, and PAM4 are digital modulations and WB-FM, AM-SSB, and AM-DSB are analog modulations. There are 20000 instances

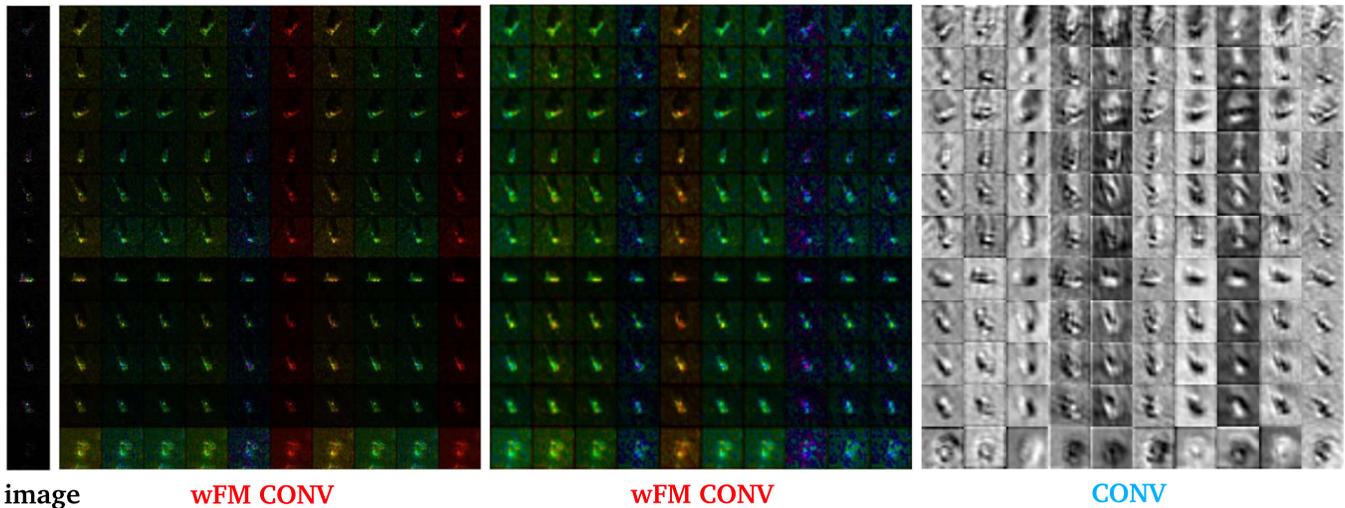


Fig. 10. Sample channel responses of our SurReal CNN on MSTAR-S images. First, there are 11 rows, with Row  $k$  containing an instance for class  $k$ . Column 1 shows the input image. Columns 2–4 contain ten channel responses in the first three convolutional layers: the first two are complex-valued wFM convolution and the last is the real-valued convolution. Second, all the complex values are displayed in a fully saturated color using the cyclic HSV colormap, with the magnitude as the intensity and the phase as the hue, whereas the real values are displayed in gray using the grayscale colormap. As the input image goes through the convolutional layers in the SurReal CNN, the distinction between instances in individual classes becomes clearer.

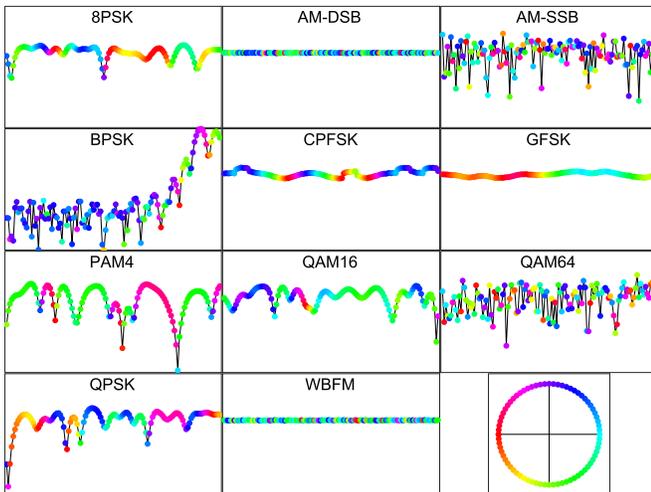


Fig. 11. RadioML sample instances. Each instance is a 128 time-step complex-valued signal at SNR of 10. We plot each signal as a line of 128 connected colored dots over time ( $x$ -axis); the height of the dot indicates the magnitude on a log scale ( $y$ -axis), and the color of the dot indicates the phase with the HSV colormap. The first 11 plots show one instance per class with the class name labeled on the top. The last plot visualizes the complex number  $\{e^{i\theta}, \theta \in [-\pi, \pi)\}$  as our colored dots, illustrating how the color varies with the phase  $\theta$ . The shape of the RF signal reflects both the message it is carrying and the modulation mode it is subject to. The classifier must ignore the distinction in the message but focus on the distinction in the modulation.

per modulation. See sample instances in Fig. 11. The data are split into 50/50 between training and testing.

2) *Real-Valued CNN Baseline*: We use O’Shea’s model [41] and feed the one-channel complex-valued RF signal as a (real and imaginary) two-channel signal.

3) *Model Size Comparison*: We follow the architecture of DCN and SurReal for images and adapt the spatial dimensions to fit the  $1 \times 128$  RF signals. Fig. 12 plots these model sizes

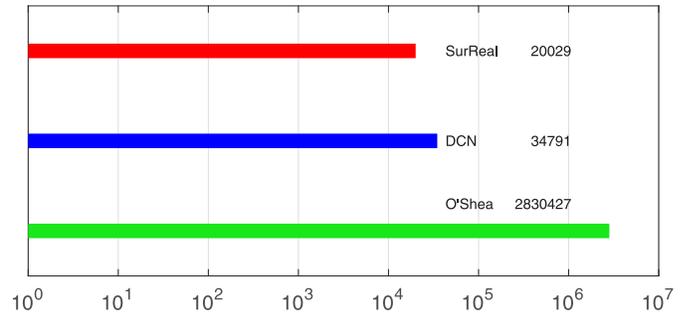


Fig. 12. Our SurReal model for RadioML is much leaner than baselines. Each model size is plotted as a horizontal bar on a log scale, labeled with the model name and the number of parameters on the right. O’Shea’s model is the real-valued baseline and the largest with 2.8M parameters. DCN is the complex-valued baseline, with 35k parameters at 1% of O’Shea. Our SurReal CNN has 20k parameters at 0.7% (58%) of O’Shea (DCN).

on a log scale. Our SurReal CNN is 0.7% of the real-valued baseline and 58% of the complex-valued baseline.

4) *Accuracy Over SNR*: Fig. 13 compares the test accuracy at various SNR levels. Our SurReal underperforms the baselines at lower SNR levels and outperforms the real-valued O’Shea baseline at higher SNRs. For example, at SNR of 10, it achieves 76.1%, surpassing O’Shea’s 72.7% with 0.7% of its model size and on par with DCN’s 76.3% at 58% of its size.

## VI. SUMMARY AND CONCLUSION

Deep learning is widely adopted in machine learning and computer vision. Most of the existing deep learning techniques are developed for data in a vector space. However, practical data often have correlations between channels and are better modeled as points a manifold.

While the Nash embedding theorem [5] assures us that it is always feasible to embed the data on a manifold into a

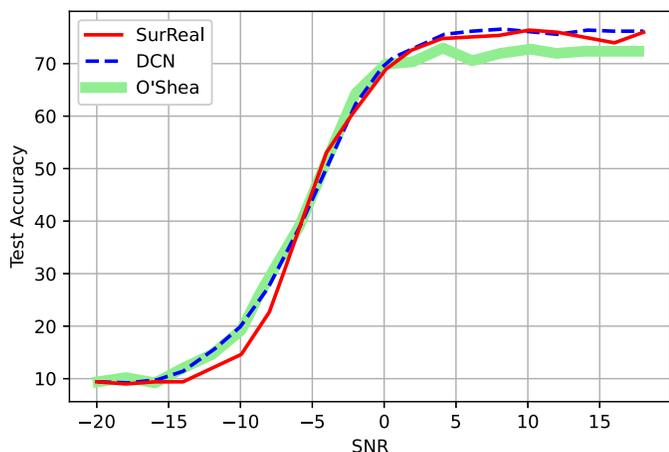


Fig. 13. Test accuracy comparison on RadioML with a varying SNR. All the models perform poorly at a very low SNR. Our SurReal underperforms the two baselines for SNR in  $[-15, -5]$ . The complex-valued baseline DCN is on par with the real-valued baseline O'Shea at a negative SNR but better at a positive SNR. Likewise, our SurReal outperforms the real-valued baseline at a positive SNR at only 0.7% (58%) size of O'Shea (DCN).

higher dimensional vector space, it would also result in an increase in the model complexity and training time. Recent geometric deep learning approaches develop tools for spaces with a certain geometry, such as graphs and surfaces.

We deal with deep learning on complex-valued data, and we approach it from a geometric perspective. The common approach is to represent complex-valued data as two-channel real-valued data, and then, all the real-valued deep learning tools can be used. However, this representation ignores the underlying geometry that defines the complex-valued data: complex-valued data containing the same information could be subject to arbitrary complex-valued scaling.

We propose to model the space of complex numbers as a product manifold of nonzero scaling and planar rotations. Arbitrary complex-valued scaling naturally becomes a group of transitive actions on this manifold. We can subsequently define convolution on the manifold that is equivariant to this action group and define distance transform that is invariant to this action group. The manifold perspective also allows us to define new nonlinear activation functions, such as tReLU and  $G$ -transport, as well as residual connections on the manifold-valued data.

A complex-valued CNN classifier composed of such layer functions has built-in invariance to complex-valued scaling so that the model only needs to focus on the discrimination between classes. Our experimental results validate our principled approach, and we dub our model SurReal based on its high performance achieved at a super-lean model size compared with real-valued or complex-valued baselines.

#### ACKNOWLEDGMENT

An earlier version of a few diagrams was constructed with help from Liu Yang. The authors would like to thank Utkarsh Singhal for thoughtful proofreading and anonymous reviewers for critical and constructive comments.

#### REFERENCES

- [1] M. Fajul Amin and K. Murase, "Single-layered complex-valued neural network for real-valued classification problems," *Neurocomputing*, vol. 72, nos. 4–6, pp. 945–955, Jan. 2009.
- [2] M. Arjovsky, A. Shah, and Y. Bengio, "Unitary evolution recurrent neural networks," in *Proc. Int. Conf. Mach. Learn.*, 2016, pp. 1120–1128.
- [3] Y. Bengio, "Learning deep architectures for AI," *Found. Trends Mach. Learn.*, vol. 2, no. 1, pp. 1–127, 2009.
- [4] N. Benvenuto and F. Piazza, "On the complex backpropagation algorithm," *IEEE Trans. Signal Process.*, vol. 40, no. 4, pp. 967–969, Apr. 1992.
- [5] W. M. Boothby, *An Introduction to Differentiable Manifolds and Riemannian Geometry*, vol. 120. New York, NY, USA: Academic, 1986.
- [6] M. M. Bronstein, J. Bruna, Y. LeCun, A. Szlam, and P. Vandergheynst, "Geometric deep learning: Going beyond Euclidean data," *IEEE Signal Process. Mag.*, vol. 34, no. 4, pp. 18–42, Jul. 2017.
- [7] J. Bruna and S. Mallat, "Invariant scattering convolution networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 8, pp. 1872–1886, Aug. 2013.
- [8] J. Bruna, W. Zaremba, A. Szlam, and Y. LeCun, "Spectral networks and locally connected networks on graphs," 2013, *arXiv:1312.6203*. [Online]. Available: <http://arxiv.org/abs/1312.6203>
- [9] J. Bruna, S. Chintala, Y. LeCun, S. Piantino, A. Szlam, and M. Tygert, "A mathematical motivation for complex-valued convolutional networks," 2015, *arXiv:1503.03438*. [Online]. Available: <http://arxiv.org/abs/1503.03438>
- [10] K. Bunte, F.-M. Schleif, and M. Biehl, "Adaptive learning for complex-valued data," in *Proc. ESANN*, 2012, pp. 1–7.
- [11] C. F. Cadieu and B. A. Olshausen, "Learning intermediate-level representations of form and motion from natural movies," *Neural Comput.*, vol. 24, no. 4, pp. 827–866, Apr. 2012.
- [12] R. Chakraborty, M. Banerjee, and B. C. Vemuri, "A CNN for homogeneous Riemannian manifolds with applications to neuroimaging," 2018, *arXiv:1805.05487*. [Online]. Available: <http://arxiv.org/abs/1805.05487>
- [13] R. Chakraborty, J. Bouza, J. Manton, and B. C. Vemuri, "ManifoldNet: A deep network framework for manifold-valued data," 2018, *arXiv:1809.06211*. [Online]. Available: <http://arxiv.org/abs/1809.06211>
- [14] R. Chakraborty *et al.*, "A statistical recurrent model on the manifold of symmetric positive definite matrices," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 8883–8894.
- [15] R. Chakraborty, J. Wang, and S. X. Yu, "Sur-real: Frechet mean and distance transform for complex-valued deep learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2019, pp. 1–9.
- [16] X. Cheng, J. He, J. He, and H. Xu, "Cv-CapsNet: Complex-valued capsule network," *IEEE Access*, vol. 7, pp. 85492–85499, 2019.
- [17] T. Cohen and M. Welling, "Group equivariant convolutional networks," in *Proc. Int. Conf. Mach. Learn.*, 2016, pp. 2990–2999.
- [18] T. Cohen, M. Geiger, J. Köhler, and M. Welling, "Convolutional networks for spherical signals," 2017, *arXiv:1709.04893*. [Online]. Available: <http://arxiv.org/abs/1709.04893>
- [19] S. Dieleman, K. W. Willett, and J. Dambre, "Rotation-invariant convolutional neural networks for galaxy morphology prediction," *Monthly Notices Roy. Astronomical Soc.*, vol. 450, no. 2, pp. 1441–1459, Jun. 2015.
- [20] D. S. Dummit and R. M. Foote, *Abstract Algebra*, vol. 3. Hoboken, NJ, USA: Wiley, 2004.
- [21] C. Esteves, C. Allen-Blanchette, X. Zhou, and K. Daniilidis, "Polar transformer networks," 2017, *arXiv:1709.01889*. [Online]. Available: <https://arxiv.org/abs/1709.01889>
- [22] M. Fréchet, *Les éléments aléatoires de Nature Quelconque Dans un Espace Distancié*, vol. 10, no. 4. Paris, France: Annales de l'institut Henri Poincaré, 1948, pp. 215–310.
- [23] W. T. Freeman and E. H. Adelson, "The design and use of steerable filters," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 13, no. 9, pp. 891–906, Sep. 1991.
- [24] G. M. Georgiou and C. Koutsougeras, "Complex domain backpropagation," *IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process.*, vol. 39, no. 5, pp. 330–334, May 1992.
- [25] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.
- [26] S. Helgason, *Differential Geometry and Symmetric Spaces*, vol. 21. New York, NY, USA: Academic, 1962.

- [27] A. Hirose and S. Yoshida, "Generalization characteristics of complex-valued feedforward neural networks in relation to signal coherence," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 23, no. 4, pp. 541–551, Apr. 2012.
- [28] E. R. Keydel, S. W. Lee, and J. T. Moore, "MSTAR extended operating conditions: A tutorial," *Proc. SPIE*, vol. 2757, pp. 228–242, Jun. 1996.
- [29] M. S. Kim and C. C. Guest, "Modification of backpropagation networks for complex-valued signal processing in frequency domain," in *Proc. IJCNN Int. Joint Conf. Neural Netw.*, Jun. 1990, pp. 27–31.
- [30] W. Kinsner and W. Grieder, "Amplification of signal features using variance fractal dimension trajectory," in *Proc. 8th IEEE Int. Conf. Cognit. Informat.*, Jun. 2009, pp. 1–17.
- [31] R. Kondor and S. Trivedi, "On the generalization of equivariance and convolution in neural networks to the action of compact groups," 2018, *arXiv:1802.03690*. [Online]. Available: <http://arxiv.org/abs/1802.03690>
- [32] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.
- [33] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.
- [34] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, p. 436, 2015.
- [35] M. Maire, T. Narihira, and S. X. Yu, "Affinity CNN: Learning pixel-centric pairwise relations for figure/ground embedding," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 174–182.
- [36] S. Mallat, "Understanding deep convolutional networks," *Philos. Trans. Roy. Soc. A, Math., Phys. Eng. Sci.*, vol. 374, no. 2065, 2016, Art. no. 20150203.
- [37] T. Needham, *Visual complex analysis*. Oxford, U.K.: Oxford Univ. Press, 1998.
- [38] T. Nitta, "On the critical points of the complex-valued neural network," in *Proc. 9th Int. Conf. Neural Inf. Process. (ICONIP)*, Nov. 2002, pp. 1099–1103.
- [39] T. Nitta, "An extension of the back-propagation algorithm to complex numbers," *Neural Netw.*, vol. 10, no. 8, pp. 1391–1415, 1997.
- [40] I. V. Oseledets, "Tensor-train decomposition," *SIAM J. Sci. Comput.*, vol. 33, no. 5, pp. 2295–2317, Jan. 2011.
- [41] T. J. O'Shea, J. Corgan, and T. C. Clancy, "Convolutional radio modulation recognition networks," in *Proc. Int. Conf. Eng. Appl. Neural Netw.* Cham, Switzerland: Springer, 2016, pp. 213–226.
- [42] T. Parcollet *et al.*, "Quaternion recurrent neural networks," 2018, *arXiv:1806.04418*. [Online]. Available: <https://arxiv.org/abs/1806.04418>
- [43] D. P. Reichert and T. Serre, "Neuronal synchrony in complex-valued deep networks," 2013, *arXiv:1312.6115*. [Online]. Available: <http://arxiv.org/abs/1312.6115>
- [44] J. Reichert, "Automatic classification of communication signals using higher order statistics," in *Proc. ICASSP IEEE Int. Conf. Acoust., Speech, Signal Process.*, vol. 5, Mar. 1992, pp. 221–224.
- [45] H. Salehian, R. Chakraborty, E. Ofori, D. Vaillancourt, and B. C. Vemuri, "An efficient recursive estimator of the Fréchet mean on a hypersphere with applications to medical image analysis," *Math. Found. Comput. Anatomy*, vol. 3, 2015.
- [46] F. Scarselli, M. Gori, A. Chung Tsoi, M. Hagenbuchner, and G. Monfardini, "The graph neural network model," *IEEE Trans. Neural Netw.*, vol. 20, no. 1, pp. 61–80, Jan. 2009.
- [47] J. Shao, C. Qu, J. Li, and S. Peng, "A lightweight convolutional neural network based on visual attention for SAR image target classification," *Sensors*, vol. 18, no. 9, p. 3039, Sep. 2018.
- [48] R. Singh, A. Chakraborty, and B. S. Manoj, "Graph Fourier transform based on directed laplacian," in *Proc. Int. Conf. Signal Process. Commun. (SPCOM)*, Jun. 2016, pp. 1–5.
- [49] C. Trabelsi *et al.*, "Deep complex networks," 2017, *arXiv:1705.09792*. [Online]. Available: <http://arxiv.org/abs/1705.09792>
- [50] P. Virtue, "Complex-valued deep learning with applications to magnetic resonance image synthesis," Ph.D. dissertation, Dept. Elect. Eng. Comput. Sci., Univ. California, Berkeley, Berkeley, CA, USA, 2018.
- [51] P. Virtue, X. Stella Yu, and M. Lustig, "Better than real: Complex-valued neural nets for MRI fingerprinting," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2017, pp. 3953–3957.
- [52] J. Wang, P. Virtue, and S. X. Yu, "Successive embedding and classification loss for aerial image classification," 2017, *arXiv:1712.01511*. [Online]. Available: <http://arxiv.org/abs/1712.01511>
- [53] D. E. Worrall, S. J. Garbin, D. Turmukhambetov, and G. J. Brostow, "Harmonic networks: Deep translation and rotation equivariance," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 5028–5037.
- [54] S. X. Yu, "Angular embedding: A robust quadratic criterion," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 1, pp. 158–173, Jan. 2012.
- [55] S. X. Yu, "Angular embedding: From jarring intensity differences to perceived luminance," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 2302–2309.
- [56] Q. Zhao, G. Zhou, S. Xie, L. Zhang, and A. Cichocki, "Tensor ring decomposition," 2016, *arXiv:1606.05535*. [Online]. Available: <http://arxiv.org/abs/1606.05535>