HyperReal: Complex-Valued Layer Functions for Complex-Valued Scaling Invariance

Anonymous authors
Paper under double-blind review

Abstract

Complex-valued measurements in MRI and SAR imaging often have complex-valued scaling ambiguity, calling for models that are invariant to complex-valued scaling of pixels. Deep Complex Networks (DCN) extends real-valued algebra to the complex domain in neural networks, but it does not address complex-valued scaling. SurReal complex-valued networks adopt a manifold view of complex numbers and derive a distance metric that is invariant to complex scaling. With distance features, it achieves complex-scaling invariance. However, rich complex-valued information is lost in this representation, and additionally, SurReal is also prevented from complex-valued non-linearity, limiting its expressive power.

We simplify the manifold formulation of SurReal and propose a new layer function that achieves complex-scaling invariance within the complex domain. We can then build hierarchical complex-valued features with complex-scaling invariance. Our so-called *HyperReal* model results in a much leaner model with better generalization. Benchmarked on MSTAR, HyperReal beats DCN (and matches SurReal) with only 3% (40%) of their respective parameter counts.

1 Introduction

Complex-valued data is a fundamental aspect of signal processing. In applications such as SAR/MR imaging and Radio Frequency communication systems, the data itself is complex-valued, but even when the data is real-valued, it is often beneficial to use complex-valued representations such as Fourier Transform and Scattering Transform (Kondor & Trivedi, 2018a; Mallat, 2016). These potential applications have motivated research on complex-valued deep learning, leading to works like Virtue (2018); Trabelsi et al. (2017), which have successfully adapted layer functions and optimization strategies from real-valued networks to the complex-valued domain.

However, this line of enquiry neglects an important practical aspect of complex-valued signal processing: Complex-valued measurements are often known only up to a complex-valued scaling constant, i.e. two complex-valued signals Z_1 and Z_2 are equivalent if and only if $Z_1 = s.Z_2$ for some scaling constant $s \in \mathbf{C}$. This equivalence motivates neural network models that are invariant to the complex-valued scaling. Given an complex-valued input image $Z \in \mathbf{C}^{H\times W}$ of size HxW, and a desired output dimension C, we define a complex-scale invariant function as follows:

Definition 1. *A function $f : \mathbf{C}^{H\times W} \to \mathbf{R}^C$ is called **complex-scale invariant** if for all inputs $\mathbf{z} \in \mathbf{C}^{H\times W}$ and scaling constants $s \in \mathbf{C}$:*

$$f(s.\mathbf{z}) = f(\mathbf{z})$$

*Similarly, it is called **complex-scale equivariant** if:*

$$f(s.\mathbf{z}) = s.f(\mathbf{z})$$

Applying these definitions to models like Deep Complex Networks (Trabelsi et al., 2017), we find that these models are not complex-scale invariant. Indeed, as shown in Fig 1, DCN is susceptible to complex-valued scaling.

This motivates the need for complex-scale invariant layers, and SurReal addresses this problem. It models complex numbers as elements of a manifold where complex-scaling corresponds to translation. In particular, its Distance Layer uses a distance metric on the manifold which is invariant to complex-scaling, and thus produces invariant features. As a result, SurReal achieves remarkable model size reduction, increased robustness, and high accuracy.

However, as we show in section 3.3, distance Layer destroys important phase information, and moreover, the previous formulation of this method cannot use complex-valued non-linear activation functions like ℂReLU. At the heart of this issue is the lack of a complex-valued complex-scale invariant layer, which is our key contribution. We recognize the algebraic structure of complex numbers, and our proposed layer divides one point with another instead of just computing distances. This operation preserves complex information that a Distance Layer cannot, and we demonstrate this effect in Section 3.3.

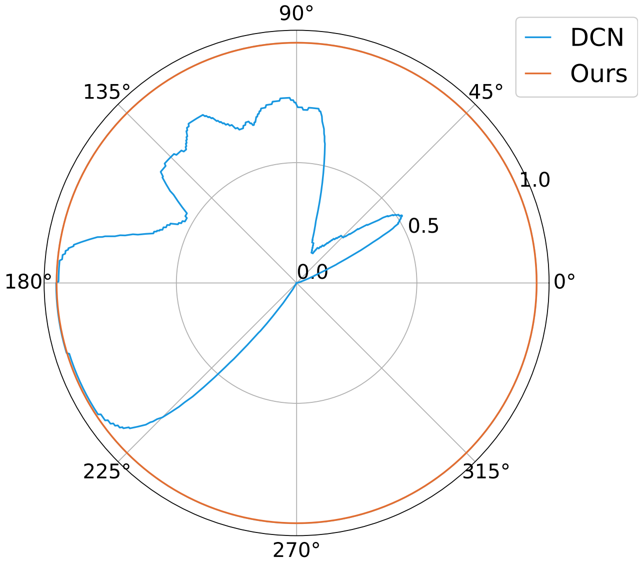

Figure 1: DCN is susceptible to complex-valued scaling, whereas our method is robust. We choose a single example, and show a polar plot of the model confidence of correct class as we multiply the image by a unit magnitude complex number; higher accuracy means larger radius. For DCN, there are three lobes of high accuracy regions, whereas our method has constant accuracy throughout the plot. This fluctuation in accuracy is due to ℂReLU breaking the complex-scale equivariance in DCN.

Our major contributions include:

- We analyze the limitations of the Distance Layer used by SurReal, and explain why SurReal architecture cannot use complex-valued non-linearities like Tangent ReLU.
- We introduce Division Layer, a new complex-valued layer which is invariant to complex scaling
- We present HyperReal, a model that matches SurReal's performance while being leaner.
- We simplify the manifold formulation of complex-scaling presented by SurReal.

2 Related Work

The field of complex numbers defines a Riemannian manifold with intrinsic geometric and algebraic properties that serve as a powerful representation of information. Given the advantages of it over other representations, complex numbers have been extensively studied in variety of fields such as mathematics, physics, and engineering (Oppenheim, 1999; Needham, 1998; Mathews & Walker, 1970).

Traditional complex-valued data analysis involves computing higher-order statistics, as seen in Kinsner & Grieder (2010); Reichert (1992). Many predecessors use the complex representation of data for modeling or embedding purposes. For instance, Cadieu & Olshausen (2012) proposes a sparse coding layer utilizing complex basis functions. Amin & Murase (2009) encodes real-valued inputs into the complex domain and proposes a classifier which forms proper boundaries for linear and

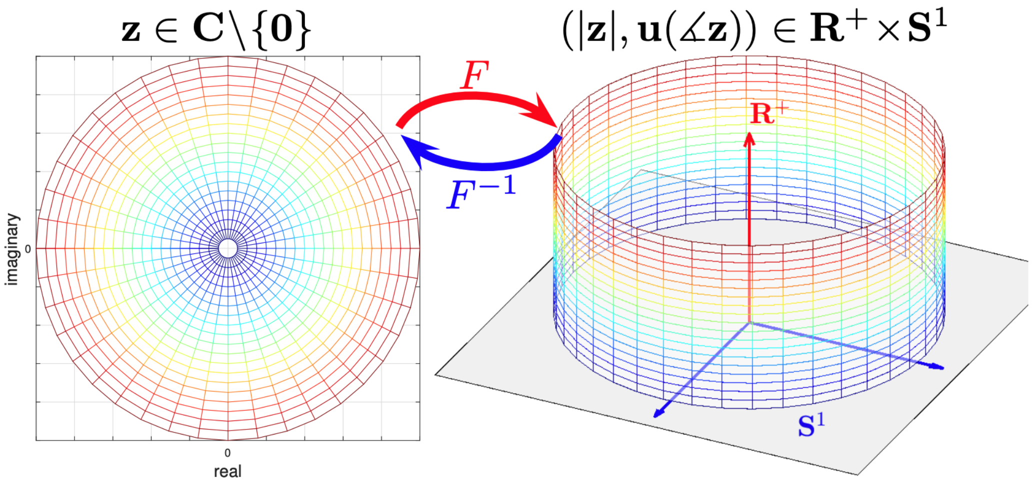
$\mathbf{z} \in \mathbf{C} \backslash \{\mathbf{0}\}$
$(|\mathbf{z}|, \mathbf{u}(\measuredangle \mathbf{z})) \in \mathbf{R}^+ \times \mathbf{S}^1$
F
F^{-1}
$\mathbf{R}^+$
$\mathbf{S}^1$
imaginary
0
0
real

Figure 2: The complex plane and the half-cylinder manifold of complex numbers. F takes a complex number, a point in the complex plane, to a 3D point on the cylinder. F^{-1} turns the 3D point back to a complex number.

non-linear problems. Reichert & Serre (2013) proposes a complex-valued model that is biologically meaningful. Yu (2009; 2012) encodes data features in a complex vector, and learns a metric system for this embedding.

More generally, complex-valued representations have found use in LSTMs and associative memory due to their robustness to noise, have been shown to have more representational power than real-valued units, and even make the optimization process easier. (Trabelsi et al., 2017; Kondor & Trivedi, 2018b; Bruna & Mallat, 2013). We refer the reader to Trabelsi et al. (2017) for a more detailed account of the benefits of complex-valued deep learning.

Contemporary progress in neural networks has been successful for real-valued data (Krizhevsky et al., 2012; He et al., 2016). Given this success, it is natural to extend the idea of neural networks to complex-valued data and to analyze the properties of complex-valued neural networks. Nitta (2002) investigates the role of critical points in the construction of a complex neural network. Hirose & Yoshida (2012) demonstrates that complex networks tend to produce smaller generalization error than real-valued network, and offers a comprehensive overview of convergence and stability of complex-valued networks.

Despite the increasing research interest in complex-valued neural networks, the problem of effectively dealing with scale-ambiguity remains open. Trabelsi et al. (2017); Zhang et al. (2017); Virtue et al. (2017); Virtue (2018) propose an extension of real neural architectures to the complex field by redefining basic building blocks such as complex convolution, batch normalization and non-linear activation function. While these methods are successful at processing artificially generated complex data (e.g. the state vector generated by as LSTM), these methods become unreliable when the input data is subject to complex-scale ambiguity. Chakraborty et al. (2019) approached this problem by adopting a manifold perspective of the problem, and our work is built on top of this line of research.

3 Methods

In order to achieve complex-scale invariance, SurReal (Chakraborty et al., 2019) models the complex number plane as a manifold where complex scaling corresponds to translation. In this section, we provide an overview of the framework, and use this framework to motivate a new type of layer, the Division Layer. For pedagogical clarity, we present a simplified yet equivalent version of the formulation used by SurReal. We refer the reader to the original paper for more details.

We start by modelling $\widetilde{\mathbf{C}} = \mathbf{C} \setminus \{0\}$, the set of non-zero complex numbers, as elements of a product manifold $\mathbf{R}^+ \times \mathbf{S}^1$. Here, $\mathbf{R}^+$ represents the magnitude component, and $\mathbf{S}^1$ is the phase component of complex rotation.

Formally, we define a bijective mapping F from $\widetilde{\mathbf{C}}$ to the manifold space $\mathbf{R}^+ \times \mathbf{S}^1$:

$$\mathbf{z} = |\mathbf{z}| \exp\left(i\, \measuredangle \mathbf{z}\right) \underset{F^{-1}}{\overset{F}{\rightleftarrows}} \left(|\mathbf{z}|, \cos(\measuredangle \mathbf{z})\ , \sin(\measuredangle \mathbf{z})\right) \tag{1}$$

This results in a cylindrical manifold where translations along the vertical axis represent increments in $\log |z|$, and thus magnitude multiplication, and the horizontal translations by θ represent multiplication by $e^{\iota\theta}$. A complex-valued input image can be considered as a point cloud on this manifold, and complex-valued scaling then becomes the translation of the entire point cloud. See figure 2. This manifold is equipped with the distance metric:

$$d\left(\mathbf{z}_1, \mathbf{z}_2\right) = \sqrt{\left(\log|\mathbf{z}_1| - \log|\mathbf{z}_2|\right)^2 + \operatorname{arc}(\measuredangle \mathbf{z_1}, \measuredangle \mathbf{z_2})^2}. \tag{2}$$

where $\operatorname{arc}(\measuredangle \mathbf{z_1}, \measuredangle \mathbf{z_2})^2$ refers to the arc-distance of the angular parts of z_1 and z_2 respectively. The structure of this distance emerges naturally from the product structure of the manifold, and the two terms in the loss function refer to distances within the vertical and horizontal components respectively. This distance function can be easily shown as invariant to complex scaling:

Proposition 1. *The distance metric defined in Eq. equation 2 is invariant to complex scaling of the inputs, i.e.:* $d(\mathbf{z_1}, \mathbf{z_2}) = \mathbf{d}(\mathbf{s}.\mathbf{z_1}, \mathbf{s}.\mathbf{z_2})$ *for any* $s \in \mathbf{C}$

Proof.

$$\begin{aligned} d(s.z_1, s.z_2) &= \sqrt{\left(\log|s\mathbf{z}_1| - \log|s\mathbf{z}_2|\right)^2 + \operatorname{arc}(\measuredangle s\mathbf{z_1}, \measuredangle s\mathbf{z_2})^2} \\ &= \sqrt{\left(\log|s| + \log|\mathbf{z}_1| - \log|s| - \log|\mathbf{z}_2|\right)^2 + \operatorname{arc}(\measuredangle s + \measuredangle \mathbf{z_1}, \measuredangle s + \measuredangle \mathbf{z_2})^2} \\ &= \sqrt{\left(\log|\mathbf{z}_1| - \log|\mathbf{z}_2|\right)^2 + \operatorname{arc}(\measuredangle \mathbf{z_1}, \measuredangle \mathbf{z_2})^2} = d(z_1, z_2) \end{aligned}$$

□

This property is the key behind SurReal's success. Unlike DCN, the invariant features extracted by SurReal allow it to be completely invariant to complex scaling, and generalize better to unseen complex-valued data with complex-scale ambiguity.

3.1 Weighted-Fréchet Mean Convolution

Since addition is not defined for arbitrary points on a manifold, we must use an alternative weighted-average operation which is compatible with manifolds: Fréchet mean. It is a generalization of weighted-averages used extensively in geometry processing literature (Panozzo et al., 2013) for smoothing curved surfaces, and is a natural substitute for Real-valued convolutions on such surfaces. wFM of a set of points (e.g. pixels) is defined as a minimizer of weighted distances. Formally:

Given $\{\mathbf{z}_i\}_{i=1}^K \subset \mathbf{C}$ are the input points and $\{w_i\}_{i=1}^K \subset (0, 1]$ with $\sum_i w_i = 1$ are the weights, (the) *weighted Fréchet mean (FM)* is defined as:

$$\text{wFM}\left(\{\mathbf{z}_i\}, \{w_i\}\right) = \arg\min_{\mathbf{m}\in\mathbf{C}} \sum_{i=1}^{K} w_i d^2\left(\mathbf{z}_i, \mathbf{m}\right). \tag{3}$$

Here, d is the distance defined in Eq. 2.

A special case is when d is the Euclidean distance: wFM reduces to the weighted average of the input points $\sum_{i=1}^K w_i z_i$. In general, however, wFM requires solving a non-convex optimization problem to locate. For the special case of the cylinder manifold, the magnitude and phase components of the wFM are decoupled, and it can be approximated easily (See Appendix A.1).

The most important property of wFM Convolution is that it equivariant both to image translations and complex-valued scaling. (See Fig 3) This allows it to be used like a regular Convolutional layer while preserving the complex-scaling properties of the model.

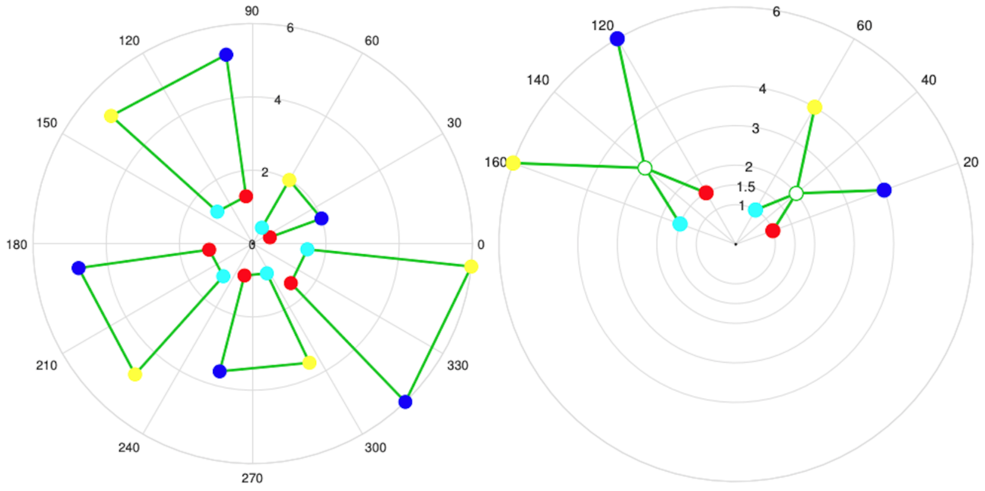

Figure 3: The convolution is equivariant to complex multiplication. a) Four colored points z_i are rotated and scaled in the complex plane. b) The wFM of the 4 points with equal weights is located at the phase midpoint and the log magnitude midpoint. It is equivariant to rotation and scaling.

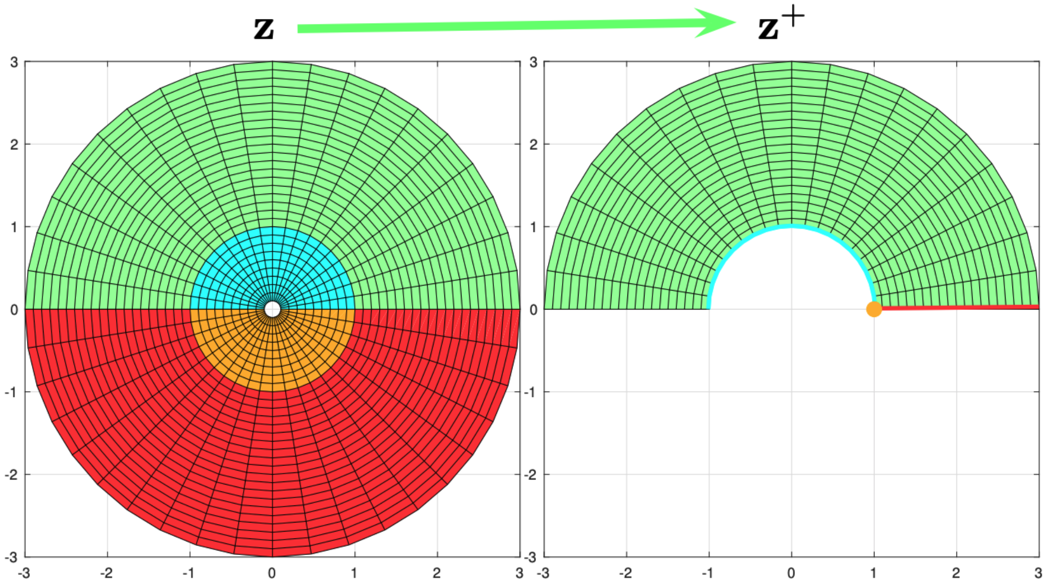

Figure 4: Here, we replicate the visualization of Tangent ReLU from Chakraborty et al. (2019). Tangent ReLU extends Euclidean ReLU to the complex plane. It divides the complex plane by the unit circle and the horizontal line: Outside the unit circle,it lets through every point in the upper half (green) and compresses the lower half to a half positive line (red). Inside the unit circle, it maps the upper half disk to the upper circle (cyan) and the lower-half disk to a single point (1,0) (brown).

3.2 Tangent ReLU

Non-linear activation functions are necessary to construct deep hierarchical representations, and ReLU is the most commonly used activation function for visual classification tasks. However, neural networks are only well-defined for vector spaces, so we adapt ReLU to this framework by applying it in the tangent space of a learned reference point. This layer is called the Tangent ReLU (TReLU), and it thresholds both phase and magnitude (See figure 4). This learned reference point is equivalent to the bias in a ReLU Network. By default, this reference point is chosen to be 1, but we learn

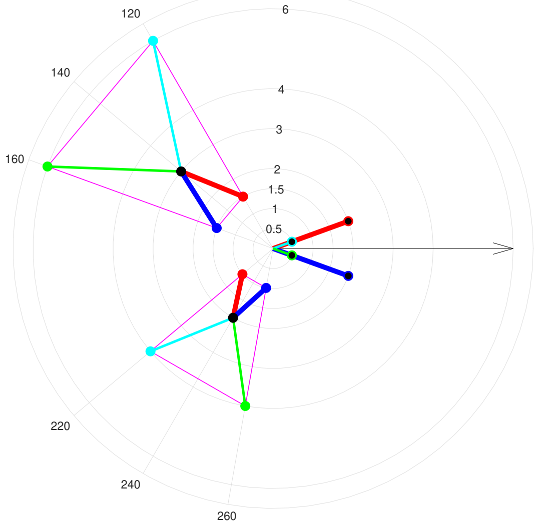

Figure 5: Our division function is invariant to complex-valued scaling. Consider 4 numbers in the complex plane, the 4 colored points on a large magenta trapezoid. Their equally weighted wFM (black dot) sits inside the trapezoid at the geometric mean of their magnitudes and the mean of their phases. When the 4 complex numbers are scaled by $\frac{2}{3}\exp(i\frac{100\pi}{180})$, the points move to the smaller trapezoid. While the manifold distances from the black dot to the colored dots are identical and invariant to scaling, the black dot divided by each of the 4 color dots results in 4 different numbers (the same colored dots with a black center around $0°$) which are also invariant to complex-valued scaling. We use this more expressive complex-valued division function instead of SurReal's real-valued distance transform to build up hierarchical complex-valued features.

it for each feature channel by multiplying each channel of the feature map by a learned complex-valued constant. This learned point is referred as the **G-Transport layer** in Chakraborty et al. (2019). Please note that while SurReal introduces Tangent ReLU as a possible non-linear activation function, their model does not use it, and as we see in Table 2, using Tangent ReLU with SurReal causes significant performance degradation. We explore the reason and solution for this failure in the next section.

3.3 Distance Layer and its Limitations

SurReal uses distances on the manifold as features. As shown in Proposition 1, the distance metric is invariant to complex-scaling, and thus the Distance Layer is the key to SurReal's complex-scale invariance. However, distance features suffer from two problems:

- Distance throws away rich information about the relative positions of different input points, and
- It makes the network incompatible with complex-valued activation functions

Firstly, Distance Layer throws away useful information by discarding all the relative phase and direction information, and only retaining the distance between the points themselves. As shown in Fig 5, it is possible to construct cases where the information is irrecoverable.

Additionally, Distance Layer produces Real-valued outputs, and thus any downstream processing is purely Real-valued. However, since complex-valued activation functions like $\mathbb{C}$ReLUand Tangent ReLU break the complex-scale equivariance, they can only be applied after an invariant layer. This contradiction prevents Surreal from using Tangent ReLU, and thus limits its complex-valued computation to be a simple linear function. Adding Tangent ReLU before the Distance layer breaks equivariance, and results in catastrophic failure of the model as shown in 2.

To fix this problem, we introduce **Division Layer**. Unlike the Distance layer which only captures distances, this layer also captures the relative positions of points. Specifically, given two complex-valued features $\mathbf{z_1}, \mathbf{z_2} \in \mathbf{C}$, we compute:

Table 1: HyperReal CNN Layer Specification

Layer Type	Input Shape	Kernel	Stride	Output Shape
wFM CONV	$[3, 1, 128, 128]$	3×3	2	$[3, 5, 63, 63]$
Division Layer	$[3, 5, 63, 63]$	3×3	1	$[3, 5, 61, 61]$
G**-transport**	$[3, 5, 61, 61]$	-	-	$[3, 5, 61, 61]$
Tangent ReLU	$[3, 5, 61, 61]$	-	-	$[3, 5, 61, 61]$
wFM CONV	$[3, 5, 61, 61]$	3×3	2	$[3, 5, 30, 30]$
Division Layer	$[3, 5, 30, 30]$	3×3	1	$[3, 5, 28, 28]$
G**-transport**	$[3, 5, 28, 28]$	-	-	$[3, 5, 28, 28]$
Tangent ReLU	$[3, 5, 28, 28]$	-	-	$[3, 5, 28, 28]$
Reshape	$[3, 5, 28, 28]$	-	-	$[15, 28, 28]$
CONV (Groups=5)	$[15, 28, 28]$	5×5	1	$[30, 24, 24]$
BN+ReLU	$[30, 24, 24]$	-	-	$[30, 24, 24]$
ResBlock	$[30, 24, 24]$	2×2	2	$[40, 24, 24]$
MaxPool	$[40, 24, 24]$	2×2	2	$[40, 12, 12]$
CONV (Groups=5)	$[40, 12, 12]$	5×5	3	$[50, 3, 3]$
BN+ReLU	$[50, 3, 3]$	-	-	$[50, 3, 3]$
ResBlock	$[50, 3, 3]$	-	-	$[60, 3, 3]$
CONV (Groups=5)	$[60, 3, 3]$	2×2	1	$[70, 2, 2]$
BN+ReLU	$[70, 2, 2]$	-	-	$[70, 2, 2]$
Average Pooling	$[70, 2, 2]$	-	-	$[70, 1, 1]$
FC	$[70]$	-	-	$[30]$
FC	$[30]$	-	-	$[10]$

Method	Parameters	20% Train	40% Train	60% Train	80% Train	100% Train
Surreal (Reproduction)	67,325	75.57 ± 2.56	88.072 ± 2.79	87.47 ± 9.31	94.65 ± 1.92	96.07 ± 0.56
Surreal + TReLU (Reproduction)	67,325	48.67 ± 4.49	51.29 ± 0.18	41.61 ± 21.23	41.27 ± 17.45	48.52 ± 3.51
HyperNet (ours)	28,695	66.05 ± 12.08	85.74 ± 1.79	89.47 ± 7.35	93.26 ± 2.99	93.24 ± 2.85
DCN	863,587	50.00 ± 2.95	57.36 ± 8.71	83.69 ± 2.29	86.53 ± 4.38	89.09 ± 1.33
3-Channel Real CNN	28,335	39.81 ± 4.02	67.17 ± 2.80	70.85 ± 2.15	78.96 ± 0.79	77.51 ± 4.36

Table 2: Our method achieves similar accuracy to SurReal on most of the training data splits. The table shows overall test accuracy on the MSTAR dataset while training on different portions of the training set. The columns referring to ”X% Train” describe the mean test accuracy and standard deviation computed over 5 different runs. SurReal+TReLU suffers catastrophic failure, but HyperNet, which combines these elements with the Division Layer, is able to converge to a good solution.

$$\text{Div}(\mathbf{z_1}, \mathbf{z_2}) = \frac{\mathbf{z_1}}{\mathbf{z_2}} \Rightarrow \big(\frac{|\mathbf{z_1}|}{|\mathbf{z_2}|}, \cos(\measuredangle \mathbf{z_1} - \measuredangle \mathbf{z_2}), \sin(\measuredangle \mathbf{z_1} - \measuredangle \mathbf{z_2})\big)$$

As shown in Fig 5, it allows us to capture more information than Distance Layer. Additionally, since the output is Complex-valued and invariant, it is possible to use complex activation functions in this architecture. This allows us to finally build deep hierarchical representations, which helps us create leaner models. In Table 2, we show that adding Tangent ReLU to SurReal leads to failure, but using a Division layer prevents this problem.

4 EXPERIMENTS

Replicating SurReal: We replicated the architecture described in Table 1. Since the paper does not mention the learning rate used to train the model, we use the same learning rate and batch size as ours, and optimize it for 200,000 iterations. We validate every 10,000 iterations, and pick the model with best validation accuracy.

Replicating DCN: We used the code publicly provided by the authors [1]. We used their image classification model with the non-linearity $\mathbb{C}$ReLU because it showed the best performance in their image classification experiments. By default, this model only accepts 32x32 images, so appended two layers of ComplexConv and ComplexBatchNorm with stride 2 to downsample the input from 128x128 to 32x32. The model was trained for 200 epochs using the schedule provided in Trabelsi et al. (2017), and the model with best validation accuracy was picked.

HyperReal: Our architecture is roughly based on the SurReal architecture from Chakraborty et al. (2019): We use a few complex-valued layers followed by a real-valued post-processor. Specifically, we use the architecture described in Table 1. The main contrast between our architecture and SurReal is the use of Division layer and Tangent ReLU. Our complex-valued processing consists of two blocks, each containing a wFM convolution, Division Layer, G-Transport, and Tangent ReLU.

Training: We optimize both SurReal and HyperReal models using Adam with learning rate 10^{-3} and batch size 64 for 2×10^5 iterations. For each setting, this process is repeated 5 times, each with a different random seed, for every configuration. We collect test accuracy statistics and report them in 2.

MSTAR: It has a total of 15,716 complex-valued X-band SAR images divided into 11 classes. We discard the last ”clutter” class containing only background, and focus only on target class recognition. See Table 2 for information about class distributions.

Preprocessing: We take the 128×128 center crop before applying method-specific pre-processing. For our models and SurReal, we normalize the magnitude and convert the data into a 3-channel (Log Magnitude, cos(phase), sin(phase)) representation. For DCN, we use the public implementation provided by the authors. This pipeline removes the pixel mean for each pixel before feeding it into the network.

Training sets: We follow the procedure of Wang et al. (2019), training on the depression angle 17 and testing on the depression angle 15. In order to test the generalization capabilities of the models, we train on several differently-sized subsets of the original training set: We train models on 20%, 40%, 60%, 80%, and full 100% of the training data respectively. Table 2 shows the results.

Results: We find that Division Layer is an effective method of combining a non-linear activation function like Tangent ReLU with the SurReal architecture. In particular, it seems to achieve excellent performance while using less than half the parameters as those used by SurReal for most training data splits. Another interesting observation is that DCN should in principle have access to a larger set of algebraic tools and a significantly higher number of parameters. However, even with this advantage, it fails on this real-world SAR data. A final noteworthy observation is that note that the entire Complex-Valued processing part of HyperReal has a total of 360 parameters compared to $\geq 28,000$ parameters in the Real-valued post-processing pipeline, but the correct inductive prior makes a big difference in generalization performance regardless of the number of parameters.

Conclusion: These experiments shed light on a previously neglected problem of complex-scale invariance, and show that it naturally comes up when working with real-world signals. We contextualize this problem by explaining it from the manifold point-of-view, and we describe how different types of approaches fail at this problem. Finally, we introduced a complex-valued invariant layer to get the best of both world. This layer shows promise in building deep hierarchical complex-valued networks, and opens up interesting possibility of a general framework to impose invariance and equivariance to other kinds of common transformations.

[1] https://github.com/ChihebTrabelsi/deep_complex_networks

References

Md Faijul Amin and Kazuyuki Murase. Single-layered complex-valued neural network for real-valued classification problems. *Neurocomputing*, 72(4-6):945–955, 2009.

Joan Bruna and Stephane Mallat. Invariant scattering convolution networks. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1872–1886, 2013.

Charles F Cadieu and Bruno A Olshausen. Learning intermediate-level representations of form and motion from natural movies. *Neural computation*, 24(4):827–866, 2012.

Rudrasis Chakraborty, Yifei Xing, and Stella Yu. Surreal: Complex-valued deep learning as principled transformations on a rotational lie group. *arXiv preprint arXiv:1910.11334*, 2019.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

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

Witold Kinsner and Warren Grieder. Amplification of signal features using variance fractal dimension trajectory. *Int. J. Cogn. Inform. Nat. Intell.*, pp. 1–17, 2010.

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*, 2018a.

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*, 2018b.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pp. 1097–1105, 2012.

Stéphane Mallat. Understanding Deep Convolutional Networks. *Philosophical Transactions A*, 374: 20150203, 2016. ISSN 1364503X. doi: 10.1098/rsta.2015.0203. URL `http://arxiv.org/abs/1601.04920`.

Jon Mathews and Robert Lee Walker. *Mathematical methods of physics*, volume 501. WA Benjamin New York, 1970.

Tristan Needham. *Visual complex analysis*. Oxford University Press, 1998.

T Nitta. On the critical points of the complex-valued neural network. In *Proceedings of the 9th International Conference on Neural Information Processing, 2002. ICONIP'02.*, volume 3, pp. 1099–1103. IEEE, 2002.

Alan V Oppenheim. *Discrete-time signal processing*. Pearson Education India, 1999.

Daniele Panozzo, Ilya Baran, Olga Diamanti, and Olga Sorkine-Hornung. Weighted averages on surfaces. *ACM Transactions on Graphics (proceedings of ACM SIGGRAPH)*, 32(4):60:1–60:12, 2013.

David P Reichert and Thomas Serre. Neuronal synchrony in complex-valued deep networks. *arXiv preprint arXiv:1312.6115*, 2013.

Juergen Reichert. Automatic classification of communication signals using higher order statistics. In *[Proceedings] ICASSP-92: 1992 IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 5, pp. 221–224. IEEE, 1992.

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.

Patrick Virtue. *Complex-valued Deep Learning with Applications to Magnetic Resonance Image Synthesis*. PhD thesis, UC Berkeley, 2018.

Patrick Virtue, Stella X. Yu, and Michael Lustig. Better than real: Complex-valued neural networks for mri fingerprinting. In *International Conference on Image Processing*, 2017.

Jiayun Wang, Patrick Virtue, and Stella Yu. Successive embedding and classification loss for aerial image classification, 2019.

Stella X. Yu. Angular embedding: from jarring intensity differences to perceived luminance. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2302–9, 2009.

Stella X. Yu. Angular embedding: A robust quadratic criterion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(1):158–73, 2012.

Zhimian Zhang, Haipeng Wang, Feng Xu, and Ya-Qiu Jin. Complex-valued convolutional neural network and its application in polarimetric sar image classification. *IEEE Transactions on Geoscience and Remote Sensing*, 55(12):7177–7188, 2017.

A APPENDIX

A.1 wFM COMPUTATION

In this section, we discuss the problem of computing the weighted-Frechet Mean for complex-valued neural networks. The goal is to solve a non-convex optimization problem to find the minimum of a weighted sum of distances. In specific cases (like Euclidean distance metric), closed form solutions exist, but there are no closed form solutions for the general case.

Given $\{\mathbf{z}_i\}_{i=1}^K \subset \mathbf{C}$ and $\{w_i\}_{i=1}^K \subset (0, 1]$ with $\sum_i w_i = 1$ and $b \in \mathbf{R}$, we aspire to compute:

$$\arg\min_{\mathbf{m}\in\mathbf{C}} \sum_{i=1}^{K} w_i d^2\left(\mathbf{z}_i, \mathbf{m}\right). \tag{4}$$

Plugging in the distance metric defined in 2, the expression reduces to:

$$\arg\min_{\mathbf{m}\in\mathbf{C}} \sum_{i=1}^{K} w_i \left((\log|\mathbf{z}_i| - \log|\mathbf{m}|)^2 + \operatorname{arc}^2(\mathbf{z_i}, \mathbf{m})\right). \tag{5}$$

Note that the objective is a sum of two terms: $(\log|\mathbf{z}_i| - \log|\mathbf{m}|)^2$ and $\operatorname{arc}^2(\mathbf{z_i}, \mathbf{m})$, where the first objective depends only on the magnitude of $\mathbf{m}$, and the second objective depends only on the phase of $\mathbf{m}$. Thus, the magnitude and phase can be solved independently of each other. The first term has a simple closed form solution, since it can be reduced to a least squares solution through a dummy variable.

$$r^* = \arg\min_{r\in\mathbf{R}} \sum_{i=1}^{K} w_i (\log|\mathbf{z}_i| - r)^2$$

$$\implies r^* = \sum_{i=1}^{K} w_i . \log|\mathbf{z}_i|$$

$$\implies \log|m^*| = \sum_{i=1}^{K} w_i . \log|\mathbf{z}_i|$$

The second term, however, is more involved. Minimizing the sum of squared arc-distances is actually a non-convex optimization problem with unit circle as its domain. Although this problem can

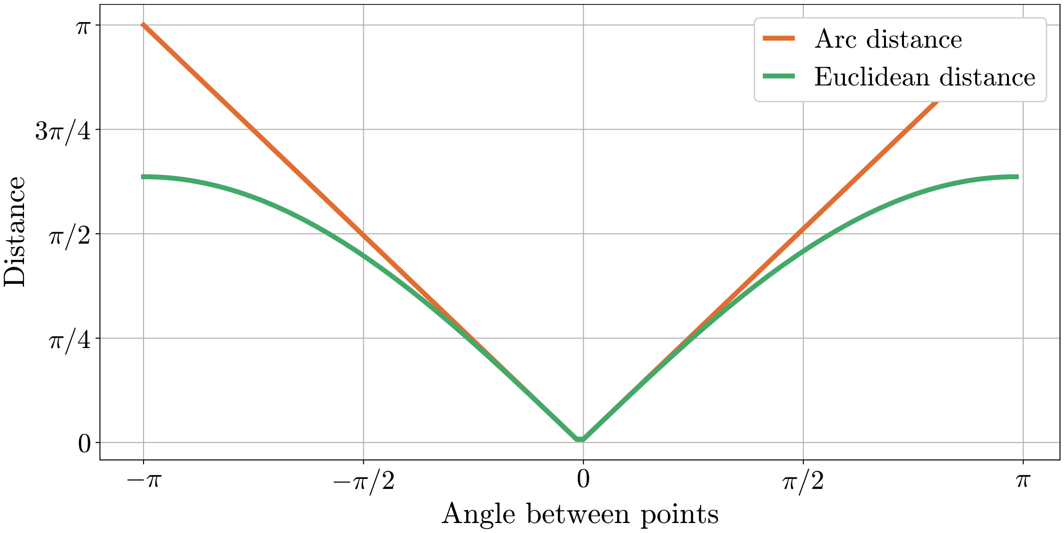

Figure 6: Different distance metrics as a function of the angle between two points. We see that the two metrics are indistinguishable for points with small angular differences, and at large angular distances, Euclidean distance is under-estimates Arc distance. Thus, for small neighborhoods with small angular variation, euclidean distance metric acts as an approximation of arc distance metric.

be solved through non-convex optimization techniques, we use a simple approximation to compute this at practical speeds: Instead of using the arc distance metric, we use the euclidean distance metric. These two metrics are closely related, but wFM on euclidean distance can be performed by simply taking a weighted average and then projecting the result onto the desired surface (Panozzo et al., 2013). This provides an efficient approximation in the form of computing a weighted average vector, and projecting it to the unit circle.

This approximation relies on euclidean distance being a faithful approximation of arc distance. Indeed, we observe that if the input points are clustered together (which is often the case for image patches), this approximation yields significantly more accurate results than for a wide support distribution. In Figure 6, we show how the two metrics vary as a function of angle between the target points.