Co-Domain Symmetry for Complex-Valued Deep Learning

Utkarsh Singhal       Yifei Xing       Stella X. Yu
An Image is a Function from Domain to Co-Domain

Domain: Pixel Locations
Co-Domain: Pixel Values
An Image is a Function from Domain to Co-Domain
Domain Transformations Act on the Pixel Coordinates

Domain Transformation:
- Translation
- Scaling
- Rotation
Domain Transformations Act on the Pixel Coordinates

- Translation
- Scaling
- Rotation

CNN [1]
Scale-Invariant CNN [2]
E(2)-Steerable CNN [3]

[1]: LeCun et al., Backpropagation Applied to Handwritten Zip Code Recognition
[2]: Xu et al., Scale-Invariance Convolutional Neural Network
[3]: Weiler et al., General E(2)-Equivariant Steerable CNNs
Co-Domain Transformations Act on the Pixel Values

Domain Transformation

- Translation
- Scaling
- Rotation

Co-domain transformation

CNN [1]
Scale-Invariant CNN [2]
E(2)-Steerable CNN [3]
Co-Domain Transformations Act on the Pixel \textit{Values}

Domain Transformation

- **translation**
  - CNN [1]

- **scaling**
  - Scale-Invariant CNN [2]

- **rotation**
  - $E(2)$-Steerable CNN [3]

Co-domain Transformation

Images: Original image of a tiger, translated, scaled, and rotated versions.
Co-Domain Encapsulates Diversity of Image Types

Thermal  
RGB  
Hyperspectral  
SAR

intensity  
color  
spectral  
complex value
We Focus on Complex-Valued Data
We CanRepresent All These Data Types in Complex Values!

Thermal | RGB | Hyperspectral | SAR |
--- | --- | --- | --- |

Complex valued encodings
Property 1: Equivalence Under Complex-Valued Scaling

\[ \hat{Z} \equiv s \cdot \hat{Z} \]
Property 2: Rich Set of Algebraic Operations

**Addition**

\[ z + w \]

**Multiplication**

\[(a + ib)(c + id) = (ac - bd) + i(ad + bc)\]

**Division**

\[
\frac{(a + bi)}{(c + di)} = \frac{ac + bd}{c^2 + d^2} + \frac{bc - ac}{c^2 + d^2}
\]

**Conjugation**

\[ z^* \]
# Methods For Handling Complex-Valued Data

<table>
<thead>
<tr>
<th>Method</th>
<th>Complex-scaling?</th>
<th>Complex-valued algebra?</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Methods For Handling Complex-Valued Data

<table>
<thead>
<tr>
<th>Method</th>
<th>Complex-scaling?</th>
<th>Complex-valued algebra?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Real-valued CNN</td>
<td>✗</td>
<td>✗</td>
</tr>
</tbody>
</table>

![Diagram showing real and imaginary parts being combined to form a real-valued CNN](image.png)
# Methods For Handling Complex-Valued Data

<table>
<thead>
<tr>
<th>Method</th>
<th>Complex-scaling?</th>
<th>Complex-valued algebra?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Real-valued CNN</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td>Deep Complex Nets</td>
<td>✗</td>
<td>✓</td>
</tr>
</tbody>
</table>

\[
W \ast h = (A \ast x - B \ast y) + i(B \ast x + A \ast y)
\]
# Methods For Handling Complex-Valued Data

<table>
<thead>
<tr>
<th>Method</th>
<th>Complex-scaling?</th>
<th>Complex-valued algebra?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Real-valued CNN</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td>Deep Complex Nets</td>
<td>✗</td>
<td>✓</td>
</tr>
<tr>
<td>SurReal</td>
<td>✓</td>
<td>✗</td>
</tr>
</tbody>
</table>

- Method: 
  - **Real-valued CNN**: Non-complex data handling, no complex-valued algebra.
  - **Deep Complex Nets**: Non-complex scaling, complex-valued algebra.
  - **SurReal**: Complex-scaling, no complex-valued algebra.

![Diagram](image.png)
## Methods For Handling Complex-Valued Data

<table>
<thead>
<tr>
<th>Method</th>
<th>Complex-scaling?</th>
<th>Complex-valued algebra?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Real-valued CNN</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td>Deep Complex Nets</td>
<td>✗</td>
<td>✓</td>
</tr>
<tr>
<td>SurReal</td>
<td>✓</td>
<td>✗</td>
</tr>
<tr>
<td><strong>Ours</strong></td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>
Benefits: Complex-Scaling Invariance

Our model makes predictions invariant to complex-valued scaling
Benefits: Higher Accuracy with Leaner Models

MSTAR: Synthetic Aperture Radar Imaging
# MSTAR: Synthetic Aperture Radar Imaging

## Benefits: Higher Accuracy with Leaner Models

<table>
<thead>
<tr>
<th>Model</th>
<th># Params</th>
<th>Relative Params</th>
<th>Training dataset size (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>5%</td>
</tr>
<tr>
<td>Complex CNN</td>
<td>863,587</td>
<td>1.00</td>
<td>49.8</td>
</tr>
<tr>
<td>Ours</td>
<td>29,536</td>
<td>0.03</td>
<td>69.5</td>
</tr>
</tbody>
</table>

Higher accuracy with much leaner models
# MSTAR: Synthetic Aperture Radar Imaging

## Model Parameters

<table>
<thead>
<tr>
<th>Model</th>
<th># Params</th>
<th>Relative Params</th>
<th>Training dataset size (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>5%</td>
</tr>
<tr>
<td>Complex CNN</td>
<td>863,587</td>
<td>1.00</td>
<td>49.8</td>
</tr>
<tr>
<td>Ours</td>
<td>29,536</td>
<td>0.03</td>
<td>69.5</td>
</tr>
</tbody>
</table>

Higher accuracy with much leaner models

Consistently lower error rate
Benefits: Diverse Filters, Lower Bias/Variance

CIFAR 10

Lower Bias and Variance
Benefits: Diverse Filters, Lower Bias/Variance

CIFAR 10

Lower Bias and Variance

Less Redundant Filters
Benefits: Robustness Against Some Types of Color Distortion

Encoding color with complex numbers
Benefits: Robustness Against Some Types of Color Distortion

Encoding color with complex numbers

![Diagram showing test accuracy percentage against color jitter for different models: Ours, Real-valued CNN, DCN, SurReal.]
Benefits: Robustness Against Some Types of Color Distortion

Encoding color with complex numbers
Methods: Complex-Scale Equi-/In-variant Layers

**Equivariant**
- Equivariant Convolution
- Equivariant Batch-Norm
- Equivariant Non-Linearity
- Equivariant Pooling

**Invariant**
- Conjugate Layer
- Division Layer
- Prototype-Distance Invariant Layer
Methods: Two Architecture Styles

Type-I

input → EConv → Division Layer → GTRelu → ×2 → Pooling → Fully Connected → Prototype Distance → output

Type-E

input → EConv → Equivariant GTRelu → ×3 → Pooling → Equivariant Fully Connected → Equivariant BN → Invariant Prototype Distance → output
Methods: Our Proposed Complex-Valued Encodings

Hyperspectral

Sliding channel encoding

\[ [x_1, x_2, x_3, x_4, \ldots] \]
\[ \rightarrow [x_1 + ix_2, x_2 + ix_3, x_3 + ix_4, \ldots] \]

Color

LAB encoding

\[ [L, a, b] \]
\[ \rightarrow [L, a + ib] \]
Thank you!

Poster 68a, June 21st, 10AM-12:30PM

github.com/sutkarsh/cds
Co-domain Symmetry for Complex-Valued Deep Learning

Utkarsh Singhal         Yifei Xing         Stella X. Yu

Contributions
1. New complex-valued learning method based on co-domain symmetry with respect to complex-valued scaling
2. New leaner classifiers with higher accuracy, better generalization, more robustness, lower model bias/variance
3. New complex-valued encodings of various types of images
4. Achieve color jitter robustness without any augmentation

Complex-Valued Data Properties

<table>
<thead>
<tr>
<th></th>
<th>Complex Algebra</th>
<th>C-scale Invariance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Real</td>
<td>✓</td>
<td>×</td>
</tr>
<tr>
<td>DCN</td>
<td>✓</td>
<td>×</td>
</tr>
<tr>
<td>SurReal</td>
<td>×</td>
<td>✓</td>
</tr>
<tr>
<td>Ours</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

Layer Functions for Co-Domain Symmetry

Equiariant

\[ f(s \cdot z) = s \cdot f(z) \]

\[
 f_{BN} = \text{BN}(|f|) \odot \frac{f}{|f|} \\
 \text{Div}(z_1, z_2) = \frac{|z_1|}{|z_2|} + \exp(i(\angle z_1 - \angle z_2)) \\
 \text{Econv}(z; W) = W \ast z = (X \ast iY) \ast (a + ib) \\
 \text{Conj}(z_1, z_2) = z_1 \overline{z_2} \\
 d(z_1, z_2) = \sqrt{(\ln|z_1| - \ln|z_2|)^2 + \arccos(\angle z_1, \angle z_2)^2} \\
 f = \text{Non-Linearity} \Rightarrow f_{out} \\
]

Invariant

\[ f(s \cdot z) = f(z) \]

Co-Domain Image Transformations

Domain: Pixel locations

Co-domain: Pixel values

Leaner, Better, More Robust Models

<table>
<thead>
<tr>
<th></th>
<th>Parameters</th>
<th>5%</th>
<th>10%</th>
<th>50%</th>
<th>100%</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSTAR</td>
<td>Real</td>
<td>33k</td>
<td>49.4</td>
<td>46.6</td>
<td>46.0</td>
</tr>
<tr>
<td>DCN</td>
<td>Real</td>
<td>36k</td>
<td>1.00</td>
<td>59.9</td>
<td></td>
</tr>
<tr>
<td>SurReal</td>
<td>DCN</td>
<td>69k</td>
<td>1.89</td>
<td>56.7</td>
<td></td>
</tr>
<tr>
<td>Ours</td>
<td>DCN</td>
<td>36k</td>
<td>0.99</td>
<td>58.7</td>
<td></td>
</tr>
</tbody>
</table>

Higher accuracy with fewer model parameters

Model Architectures: Early or Late Invariance

Type-I

input \xrightarrow{\times 3} \text{Encoder} \xrightarrow{\text{Divisive Layer}} \text{Simplified GNN} \xrightarrow{\text{Invarient Prototype Distance}} \text{Pooling} \xrightarrow{\text{Early Invariance}} \text{Output} \\

Type-E

input \xrightarrow{\text{Invarient Prototype Distance}} \text{Outer Product} \xrightarrow{\text{Pooling}} \text{Early Invariance} \xrightarrow{\text{Output}} \\

Less redundant filters

Color robustness

Lower bias/variance

Invariance