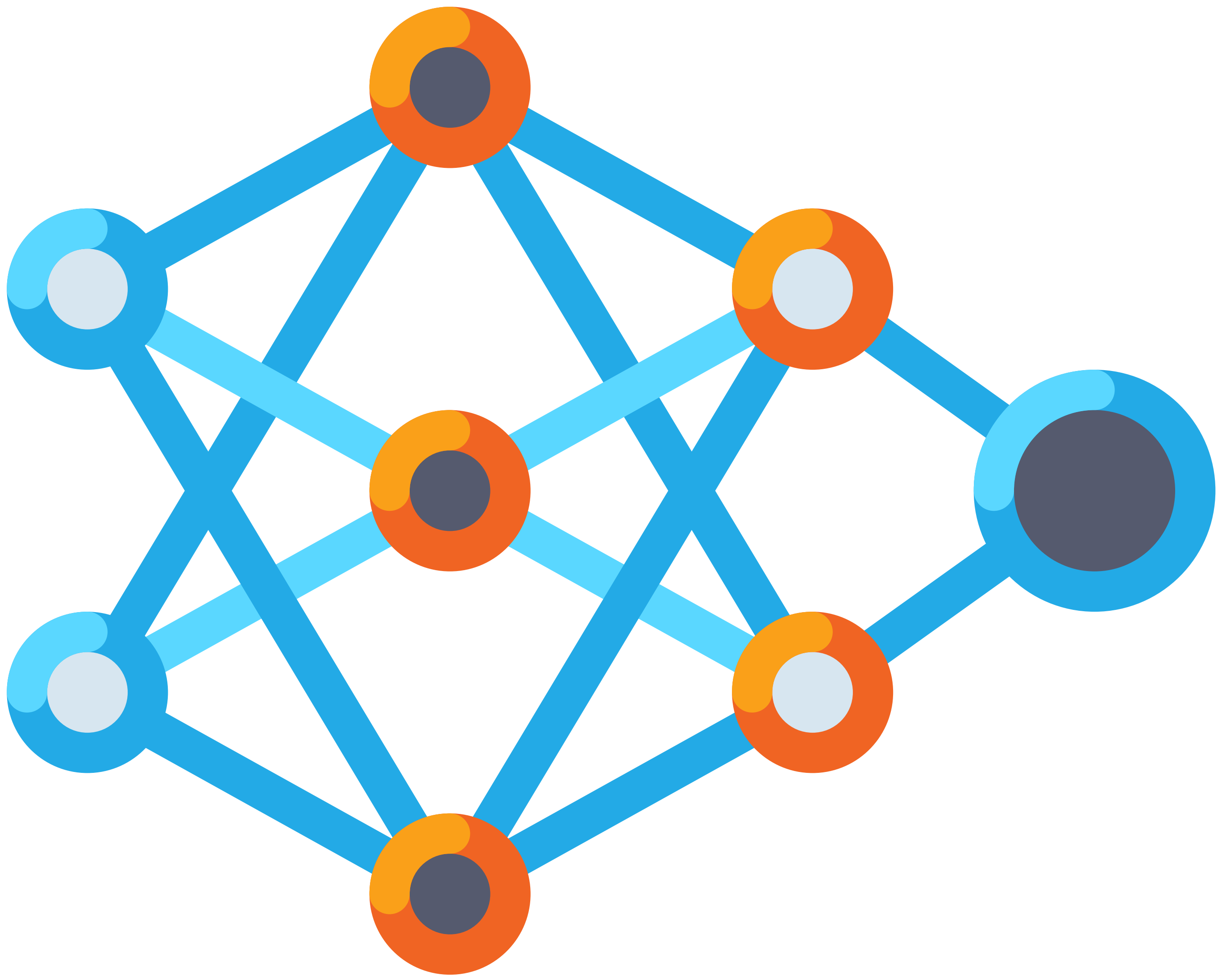
Analysis of Complex Valued Convolutional Networks

Deep Complex Networks

Source: openreview.net/forum?id=H1T2hmZAb



Mathematical Background

1. Complex Valued Convolution: Distributive property (simulating complex arithmetic using real-valued entities).

$$\mathbf{W} * \mathbf{h} = (\mathbf{A} * \mathbf{x} - \mathbf{B} * \mathbf{y}) + i\,(\mathbf{B} * \mathbf{x} + \mathbf{A} * \mathbf{y}).$$

$$\begin{bmatrix} \Re(\mathbf{W} * \mathbf{h}) \\ \Im(\mathbf{W} * \mathbf{h}) \end{bmatrix} = \begin{bmatrix} \mathbf{A} & -\mathbf{B} \\ \mathbf{B} & \mathbf{A} \end{bmatrix} * \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix}.$$

2. Complex Valued Activations

$$z\mathrm{ReLU}(z) = \begin{cases} z & \text{if } \theta_z \in [0, \pi/2], \\ 0 & \text{otherwise}, \end{cases}$$

$$\mathrm{modReLU}(z) = \mathrm{ReLU}(|z| + b)\, e^{i\theta_z} = \begin{cases} (|z| + b)\frac{z}{|z|} & \text{if } |z| + b \geq 0, \\ 0 & \text{otherwise}, \end{cases}$$

$$\mathbb{C}\mathrm{ReLU}(z) = \mathrm{ReLU}(\Re(z)) + i\,\mathrm{ReLU}(\Im(z)).$$

Mathematical Background

3. Complex Differentiability (Standard Cauchy Riemann Equations)

$$f(x+iy) = f(x, y) = u(x, y) + iv(x, y) \qquad \frac{\partial u}{\partial y} = -\frac{\partial v}{\partial x}, \quad \frac{\partial u}{\partial x} = \frac{\partial v}{\partial y}$$

$$f'(z_0) \equiv \lim_{\Delta z \to 0} \left[\frac{(f(z_0) + \Delta z) - f(z_0)}{\Delta z} \right]$$

$$= \lim_{\Delta x \to 0} \lim_{\Delta y \to 0} \left[\frac{\Delta u(x_0, y_0) + i\, \Delta v(x_0, y_0)}{\Delta x + i\, \Delta y} \right]$$

$$= \lim_{\Delta x \to 0} \left[\frac{\Delta u(x_0, y_0) + i\, \Delta v(x_0, y_0)}{\Delta x + i\, 0} \right]$$

$$= \lim_{\Delta y \to 0} \left[\frac{\Delta u(x_0, y_0) + i\, \Delta v(x_0, y_0)}{0 + i\, \Delta y} \right]$$

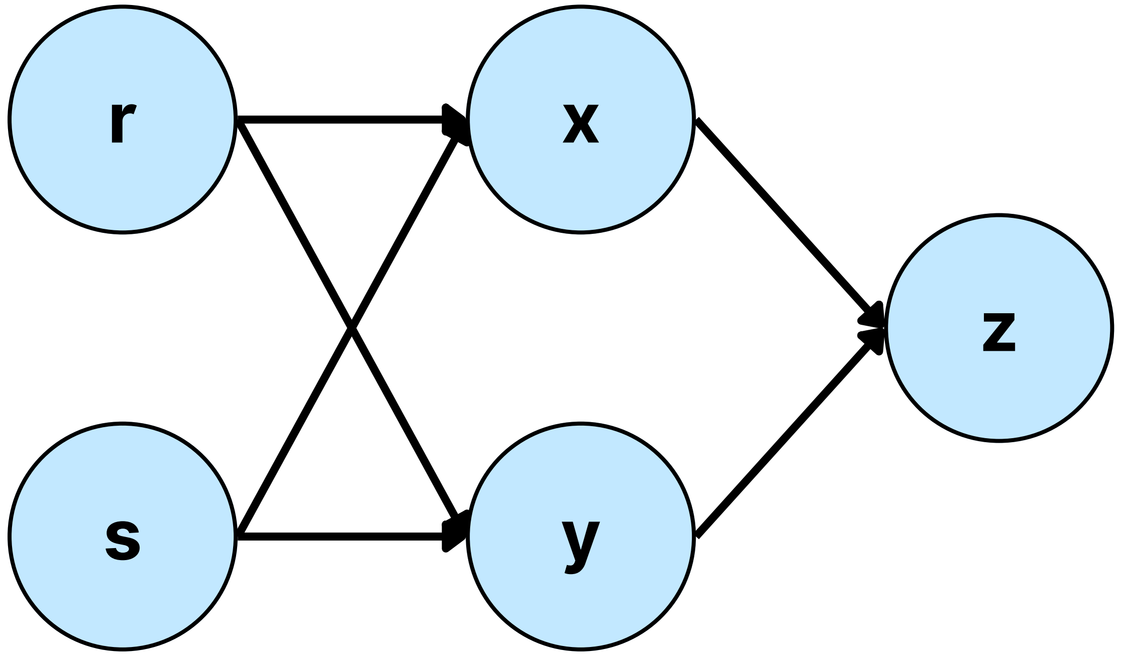
Mathematical Background

4. Backpropogation

$z = x + i\,y \quad t = r + i\,s$

$$\nabla_L(z) = \frac{\partial L}{\partial z} = \frac{\partial L}{\partial x} + i\frac{\partial L}{\partial y} = \frac{\partial L}{\partial \Re(z)} + i\frac{\partial L}{\partial \Im(z)} = \Re(\nabla_L(z)) + i\Im(\nabla_L(z))$$

$$\begin{aligned}
\nabla_L(t) &= \frac{\partial L}{\partial t} = \frac{\partial L}{\partial r} + i\frac{\partial L}{\partial s} \\
&= \frac{\partial L}{\partial x}\frac{\partial x}{\partial r} + \frac{\partial L}{\partial y}\frac{\partial y}{\partial r} + i\left(\frac{\partial L}{\partial x}\frac{\partial x}{\partial s} + \frac{\partial L}{\partial y}\frac{\partial y}{\partial s}\right) \\
&= \frac{\partial L}{\partial x}\left(\frac{\partial x}{\partial r} + i\frac{\partial x}{\partial s}\right) + \frac{\partial L}{\partial y}\left(\frac{\partial y}{\partial r} + i\frac{\partial y}{\partial s}\right) \\
&= \frac{\partial L}{\partial \Re(z)}\left(\frac{\partial x}{\partial r} + i\frac{\partial x}{\partial s}\right) + \frac{\partial L}{\partial \Im(z)}\left(\frac{\partial y}{\partial r} + i\frac{\partial y}{\partial s}\right) \\
&= \Re(\nabla_L(z))\left(\frac{\partial x}{\partial r} + i\frac{\partial x}{\partial s}\right) + \Im(\nabla_L(z))\left(\frac{\partial y}{\partial r} + i\frac{\partial y}{\partial s}\right)
\end{aligned}$$



Mathematical Background

5. Complex Batch Normalization: To ensure equal variance in both the real and imaginary components, (the resulting distribution need to be circular, not elliptical with high eccentricity)

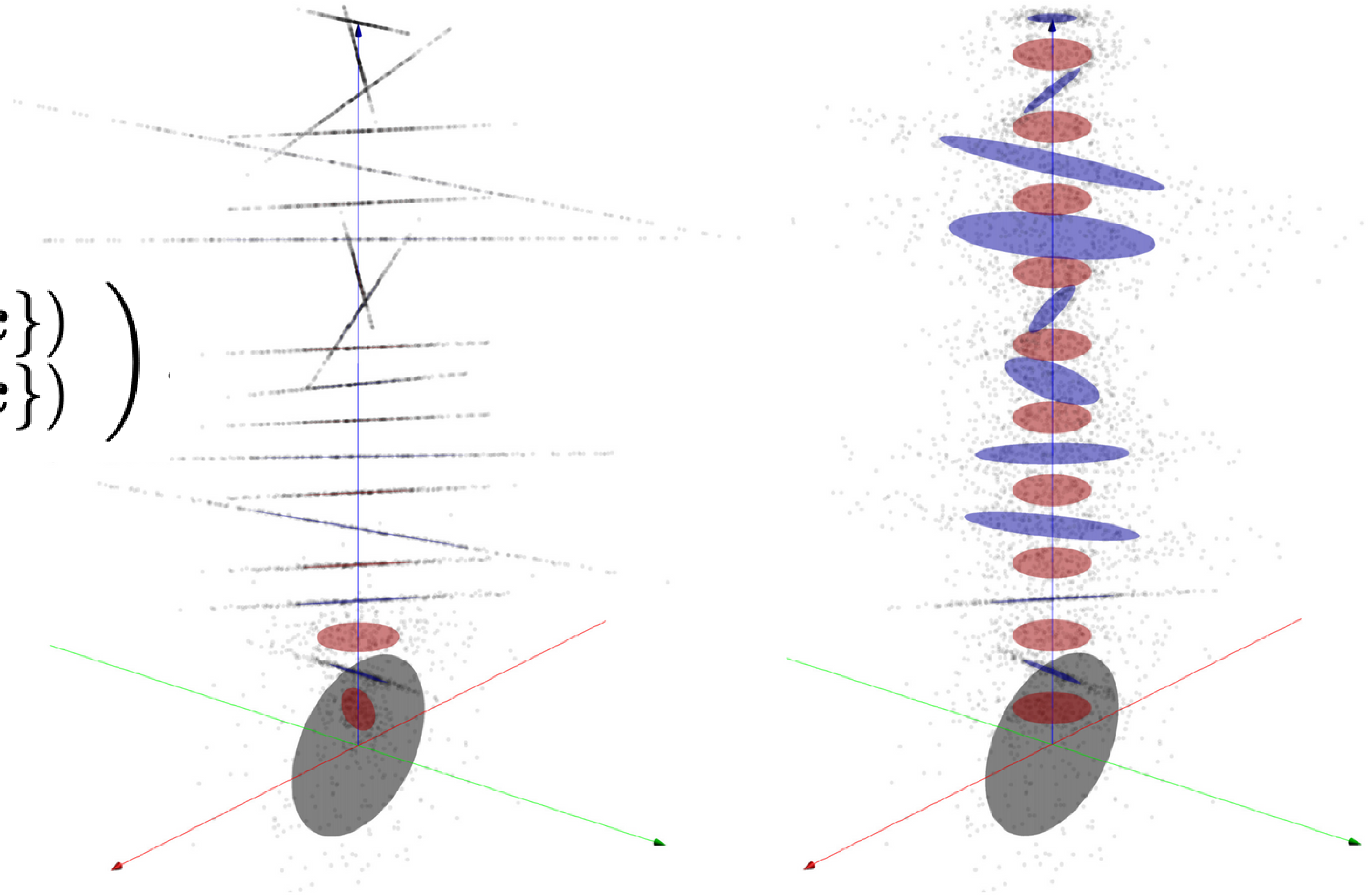
$$\tilde{\boldsymbol{x}} = (\boldsymbol{V})^{-\frac{1}{2}} (\boldsymbol{x} - \mathbb{E}[\boldsymbol{x}])$$

$$\boldsymbol{V} = \begin{pmatrix} V_{rr} & V_{ri} \\ V_{ir} & V_{ii} \end{pmatrix}$$

$$= \begin{pmatrix} \operatorname{Cov}(\Re\{\boldsymbol{x}\}, \Re\{\boldsymbol{x}\}) & \operatorname{Cov}(\Re\{\boldsymbol{x}\}, \Im\{\boldsymbol{x}\}) \\ \operatorname{Cov}(\Im\{\boldsymbol{x}\}, \Re\{\boldsymbol{x}\}) & \operatorname{Cov}(\Im\{\boldsymbol{x}\}, \Im\{\boldsymbol{x}\}) \end{pmatrix}$$

$$\operatorname{BN}(\tilde{\boldsymbol{x}}) = \boldsymbol{\gamma}\,\tilde{\boldsymbol{x}} + \boldsymbol{\beta}.$$

$$\gamma = \begin{pmatrix} \gamma_{rr} & \gamma_{ri} \\ \gamma_{ri} & \gamma_{ii} \end{pmatrix}$$



Mathematical Background

6. Weight Initialization

$$W = |W|e^{i\theta} = \Re\{W\} + i\,\Im\{W\}$$

Phase: U[-pi, pi]

Magnitude: Rayleigh(σ): sqrt(X^2 + Y^2), X and Y are normal with sigma

$$\mathrm{Var}(W) = \mathbb{E}\left[WW^*\right] - \left(\mathbb{E}\left[W\right]\right)^2 = \mathbb{E}\left[|W|^2\right] - \left(\mathbb{E}\left[W\right]\right)^2,$$

$$\mathrm{Var}(|W|) = \mathbb{E}\left[|W||W|^*\right] - \left(\mathbb{E}\left[|W|\right]\right)^2 = \mathbb{E}\left[|W|^2\right] - \left(\mathbb{E}\left[|W|\right]\right)^2$$

$$\Downarrow$$

$$\mathrm{Var}(W) = \mathrm{Var}(|W|) + \left(\mathbb{E}\left[|W|\right]\right)^2$$

$$\mathbb{E}\left[|W|\right] = \sigma\sqrt{\frac{\pi}{2}}, \quad \mathrm{Var}(|W|) = \frac{4-\pi}{2}\sigma^2 \Longrightarrow \mathrm{Var}(W) = \frac{4-\pi}{2}\sigma^2 + \left(\sigma\sqrt{\frac{\pi}{2}}\right)^2 = 2\sigma^2$$

$$\sigma = 1/\sqrt{n_{in} + n_{out}}$$

$$\sigma = 1/\sqrt{n_{in}}$$

Model Overview

$$\tilde{\boldsymbol{x}} = (\boldsymbol{V})^{-\frac{1}{2}} (\boldsymbol{x} - \mathbb{E}[\boldsymbol{x}])$$

$$\mathrm{BN}(\tilde{\boldsymbol{x}}) = \boldsymbol{\gamma} \tilde{\boldsymbol{x}} + \boldsymbol{\beta}.$$

$$BN \rightarrow ReLU \rightarrow Conv \rightarrow BN \rightarrow ReLU \rightarrow Conv$$

$$\mathbf{W} * \mathbf{h} = (\mathbf{A} * \mathbf{x} - \mathbf{B} * \mathbf{y}) + i\,(\mathbf{B} * \mathbf{x} + \mathbf{A} * \mathbf{y}).$$

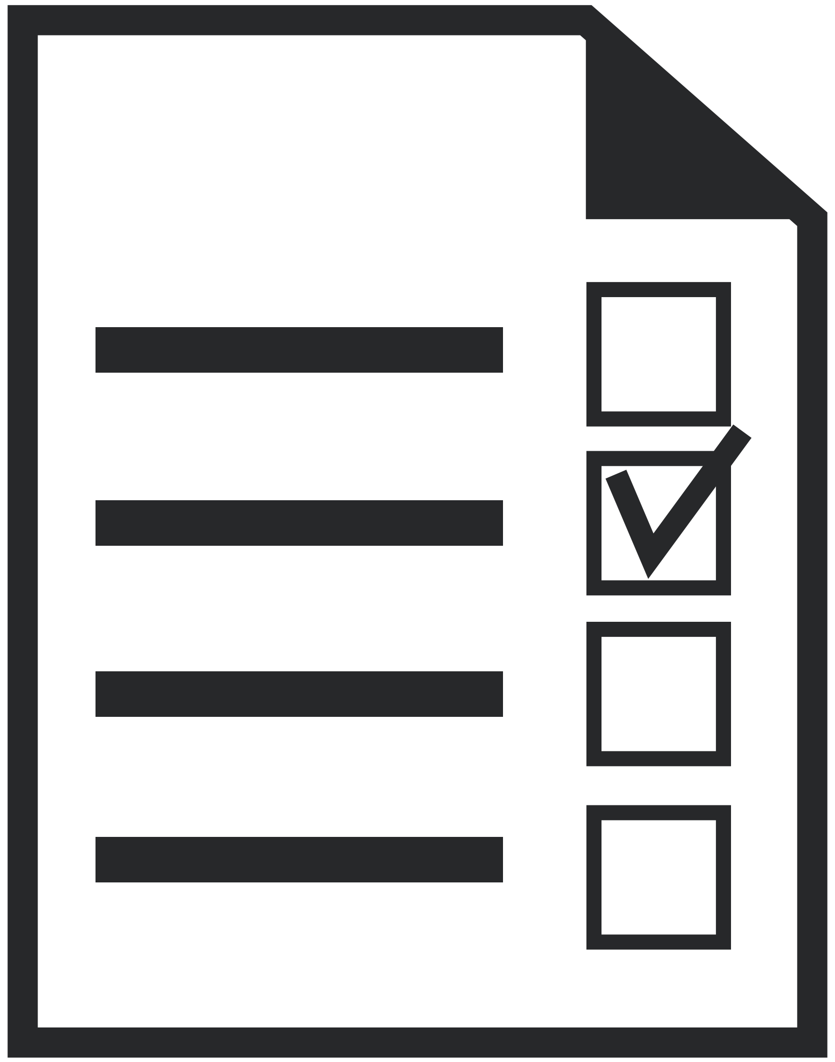
$$\begin{bmatrix} \Re(\mathbf{W} * \mathbf{h}) \\ \Im(\mathbf{W} * \mathbf{h}) \end{bmatrix} = \begin{bmatrix} \mathbf{A} & -\mathbf{B} \\ \mathbf{B} & \mathbf{A} \end{bmatrix} * \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix}.$$

$$\mathbb{C}\mathrm{ReLU}(z) = \mathrm{ReLU}(\Re(z)) + i\,\mathrm{ReLU}(\Im(z)).$$

Results I

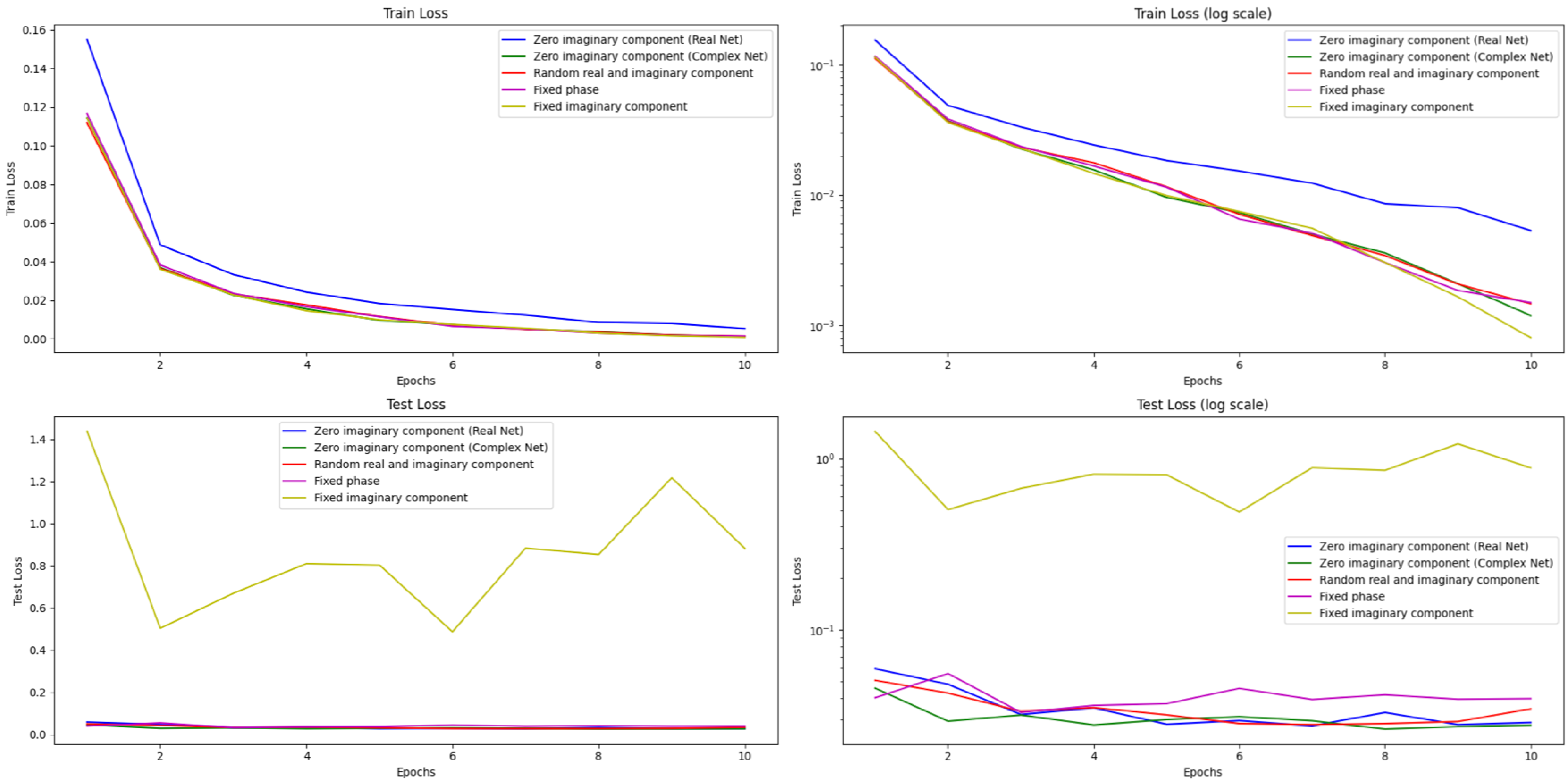
EXP1: Standard Image Datasets
Comparison with:

1. Real Net
2. Complex Net
3. Complex Net with Fixed Phase
4. Complex net with Fixed Imaginary Component
5. Complex net with Random Real and Imaginary Component

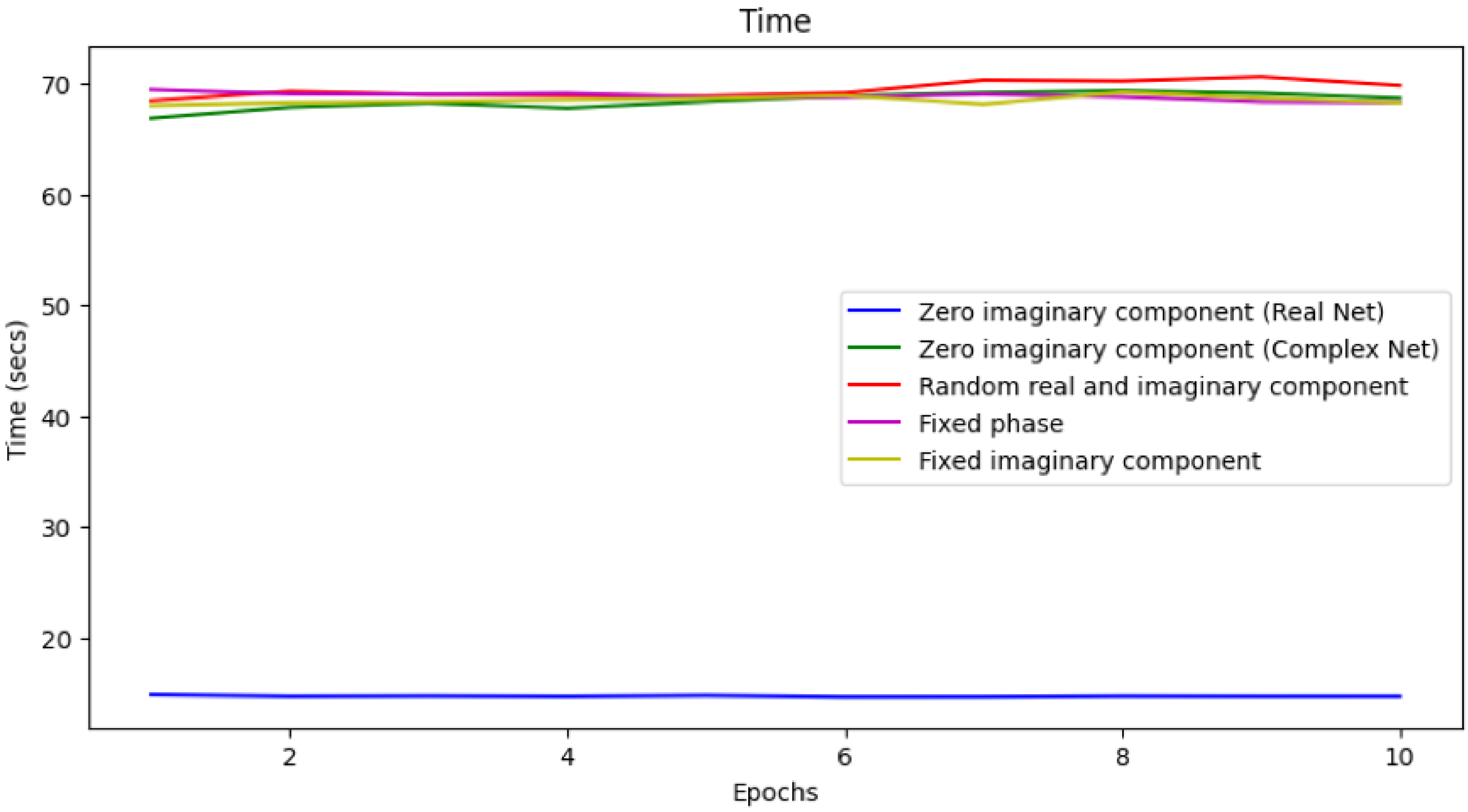


Training Settings: MNIST

```
----------------------------------------------------------------
        Layer (type)               Output Shape         Param #
================================================================
            Conv2d-1           [-1, 10, 24, 24]             260
       BatchNorm2d-2           [-1, 10, 12, 12]              20
            Conv2d-3             [-1, 20, 8, 8]           5,020
            Linear-4                  [-1, 500]         160,500
            Linear-5                   [-1, 10]           5,010
================================================================
Total params: 170,810
Trainable params: 170,810
Non-trainable params: 0
----------------------------------------------------------------
Input size (MB): 0.00
Forward/backward pass size (MB): 0.07
Params size (MB): 0.65
Estimated Total Size (MB): 0.72
----------------------------------------------------------------
```

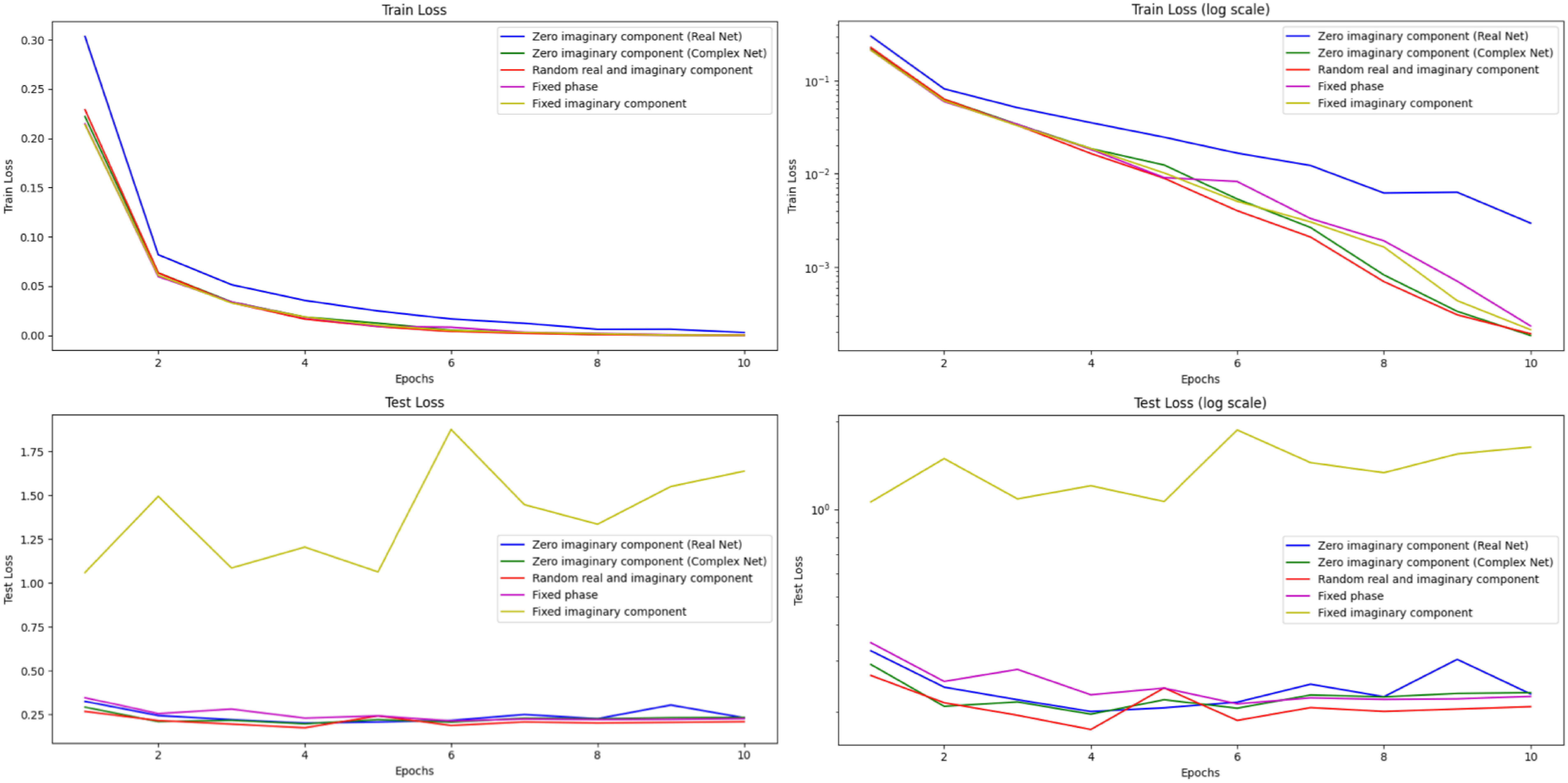


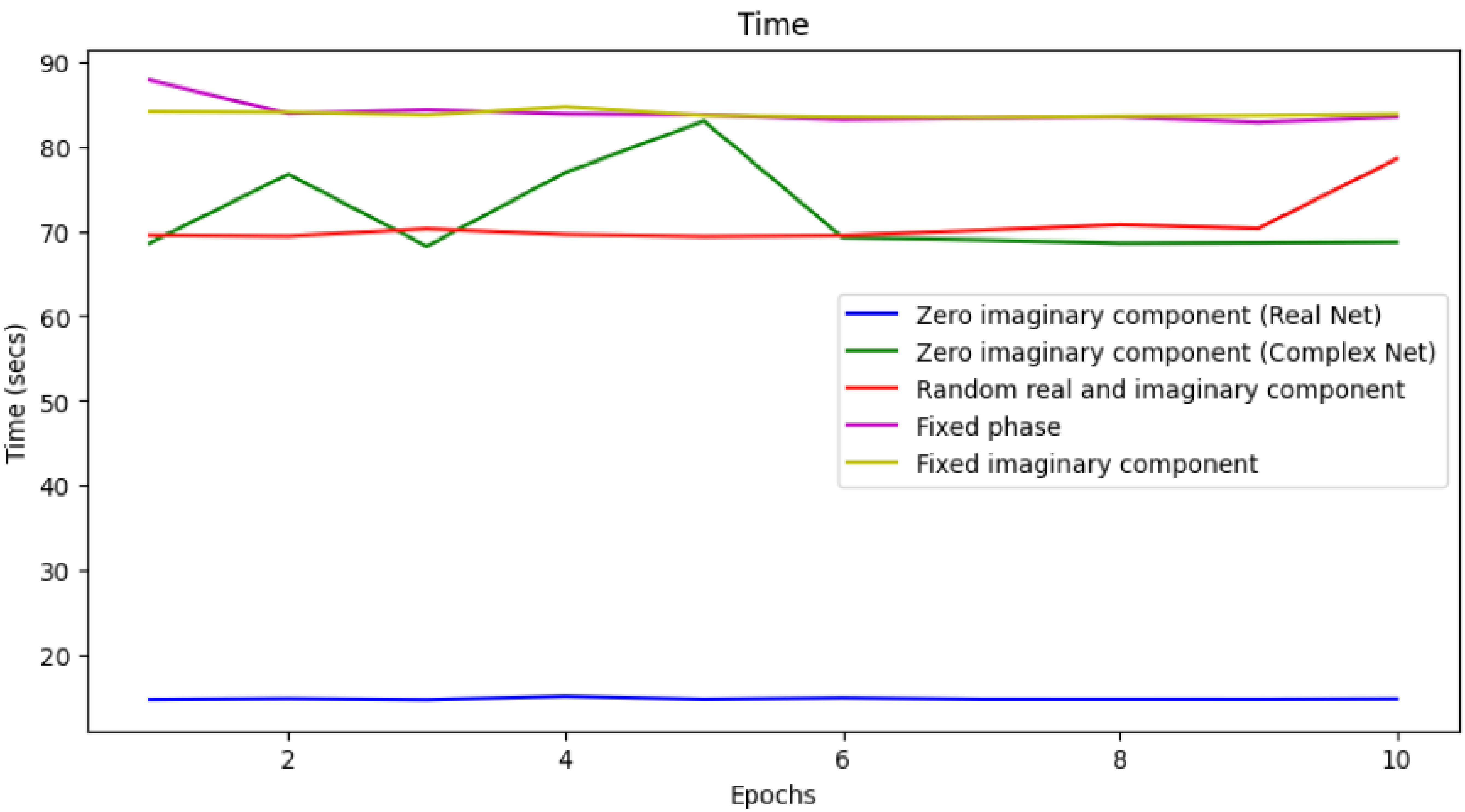
Train Loss
Train Loss (log scale)
Test Loss
Test Loss (log scale)
Train Loss
Test Loss
Epochs
Zero imaginary component (Real Net)
Zero imaginary component (Complex Net)
Random real and imaginary component
Fixed phase
Fixed imaginary component


Time
70
60
50
40
30
20
Time (secs)
2
4
6
8
10
Epochs
Zero imaginary component (Real Net)
Zero imaginary component (Complex Net)
Random real and imaginary component
Fixed phase
Fixed imaginary component

Training Settings: KMNIST

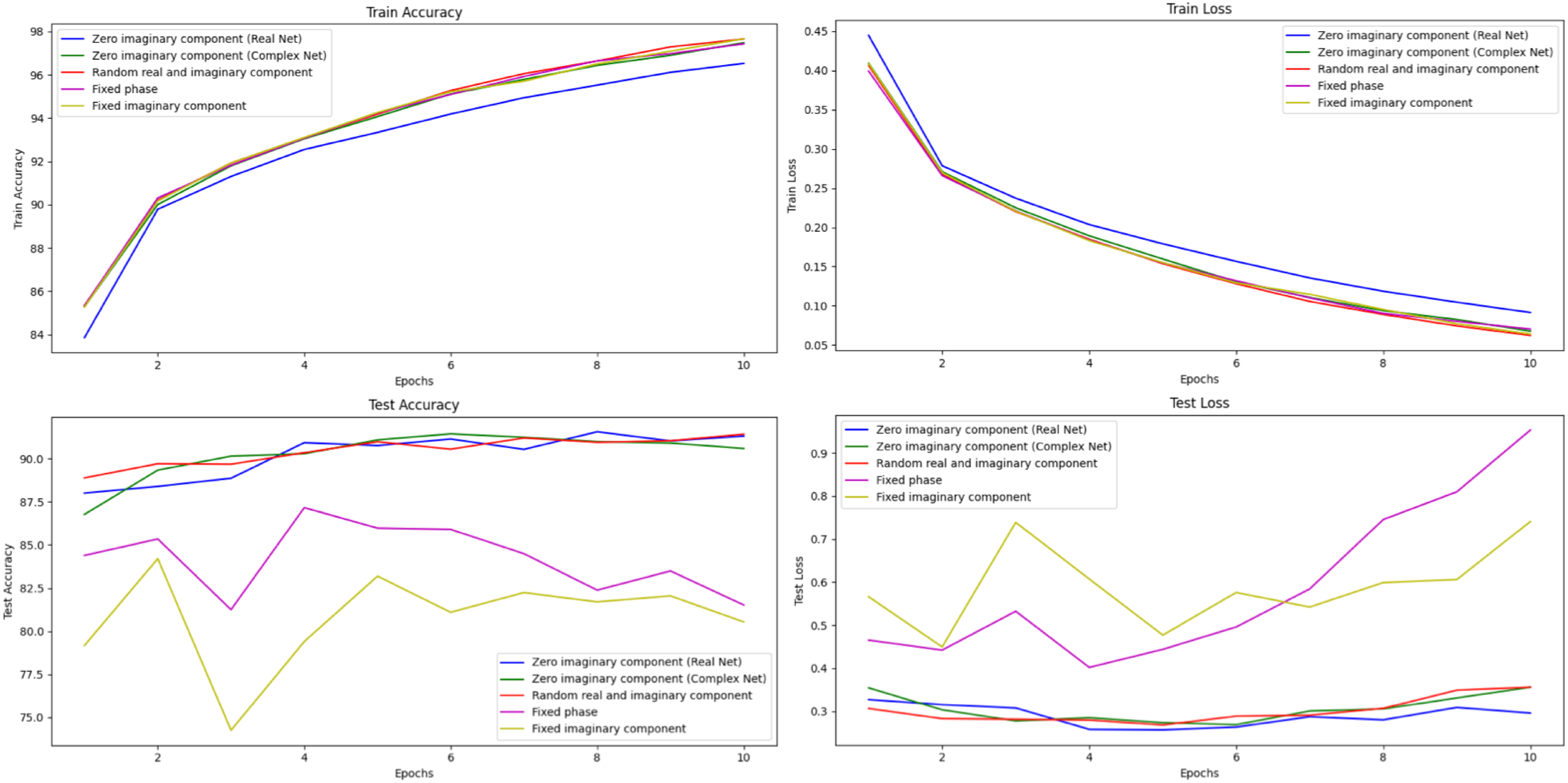
```
----------------------------------------------------------------
        Layer (type)               Output Shape         Param #
================================================================
            Conv2d-1           [-1, 10, 24, 24]             260
       BatchNorm2d-2           [-1, 10, 12, 12]              20
            Conv2d-3             [-1, 20, 8, 8]           5,020
            Linear-4                  [-1, 500]         160,500
            Linear-5                   [-1, 10]           5,010
================================================================
Total params: 170,810
Trainable params: 170,810
Non-trainable params: 0
----------------------------------------------------------------
Input size (MB): 0.00
Forward/backward pass size (MB): 0.07
Params size (MB): 0.65
Estimated Total Size (MB): 0.72
----------------------------------------------------------------
```


Train Loss
Train Loss (log scale)
Test Loss
Test Loss (log scale)
Train Loss
Test Loss
Epochs
0.30
0.25
0.20
0.15
0.10
0.05
0.00
1.75
1.50
1.25
1.00
0.75
0.50
0.25
10^{-1}
10^{-2}
10^{-3}
10^{0}
2
4
6
8
10
Zero imaginary component (Real Net)
Zero imaginary component (Complex Net)
Random real and imaginary component
Fixed phase
Fixed imaginary component

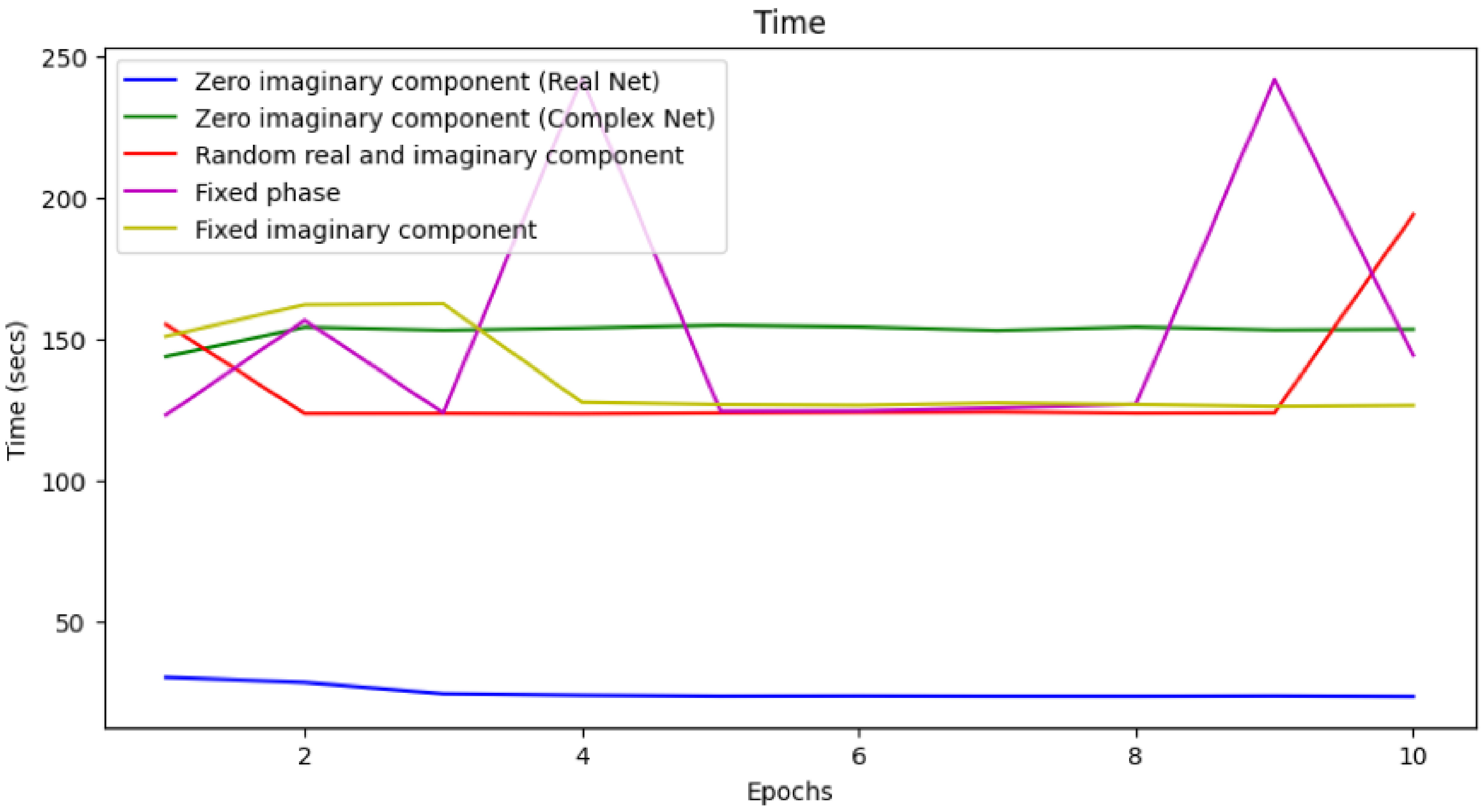

Time
Time (secs)
Epochs
90
80
70
60
50
40
30
20
2
4
6
8
10
Zero imaginary component (Real Net)
Zero imaginary component (Complex Net)
Random real and imaginary component
Fixed phase
Fixed imaginary component

Training Settings: FMNIST

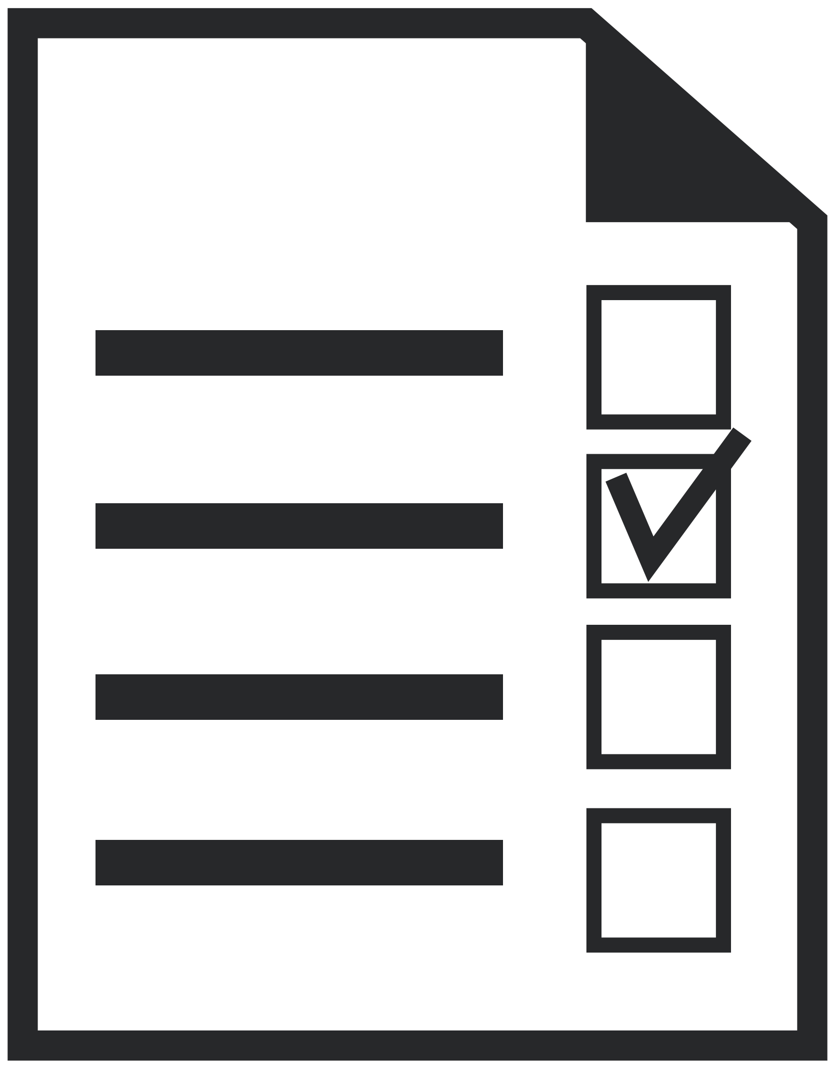
```
----------------------------------------------------------------
        Layer (type)               Output Shape         Param #
================================================================
            Conv2d-1           [-1, 32, 24, 24]             832
       BatchNorm2d-2           [-1, 32, 12, 12]              64
            Conv2d-3             [-1, 64, 8, 8]          51,264
            Linear-4                  [-1, 500]         512,500
            Linear-5                   [-1, 10]           5,010
================================================================
Total params: 569,670
Trainable params: 569,670
Non-trainable params: 0
----------------------------------------------------------------
Input size (MB): 0.00
Forward/backward pass size (MB): 0.21
Params size (MB): 2.17
Estimated Total Size (MB): 2.39
----------------------------------------------------------------
```



Train Accuracy
Zero imaginary component (Real Net)
Zero imaginary component (Complex Net)
Random real and imaginary component
Fixed phase
Fixed imaginary component
Train Accuracy
Epochs
Train Loss
Zero imaginary component (Real Net)
Zero imaginary component (Complex Net)
Random real and imaginary component
Fixed phase
Fixed imaginary component
Train Loss
Epochs
Test Accuracy
Zero imaginary component (Real Net)
Zero imaginary component (Complex Net)
Random real and imaginary component
Fixed phase
Fixed imaginary component
Test Accuracy
Epochs
Test Loss
Zero imaginary component (Real Net)
Zero imaginary component (Complex Net)
Random real and imaginary component
Fixed phase
Fixed imaginary component
Test Loss
Epochs


Time
Zero imaginary component (Real Net)
Zero imaginary component (Complex Net)
Random real and imaginary component
Fixed phase
Fixed imaginary component
250
200
150
100
50
Time (secs)
2
4
6
8
10
Epochs

Results 2



EXP2: Music Genre Dataset [Audio Signal Clasification]
Comparison with:

1. Real Valued MFCCs for Real Net
2. Real Valued MFCCs for Complex Net
3. Complex Valued MFCCs for Complex Net

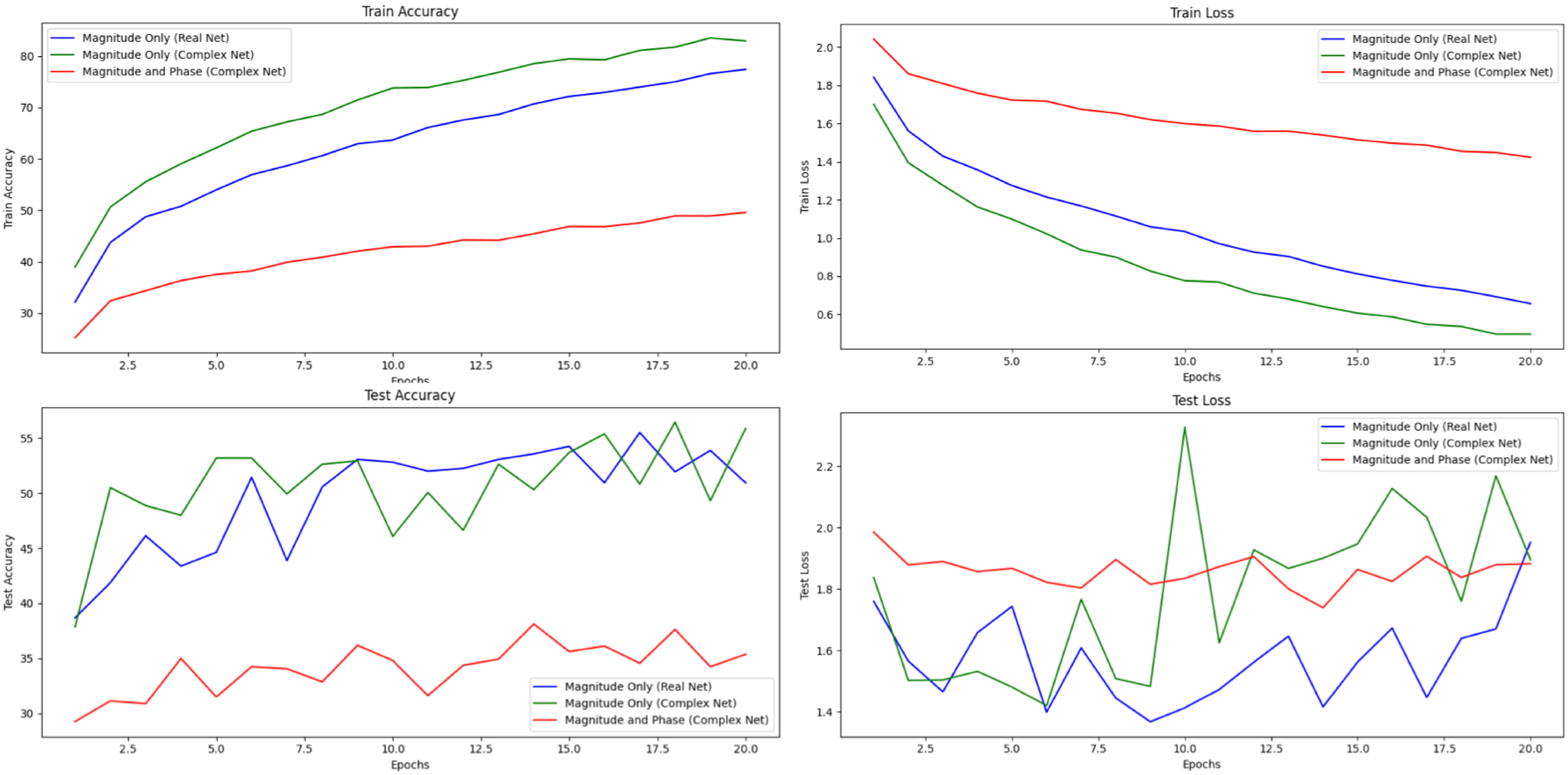
MFCC Extraction Workflow: I

1. Sampling, Normalization, Framing and Windowing
2. Discrete Fourier Transform
3. Power Spectrum Calculation
4. Mel Frequency Bin Construction
5. Log Spectrum Calculation
6. Discrete Cosine Transform

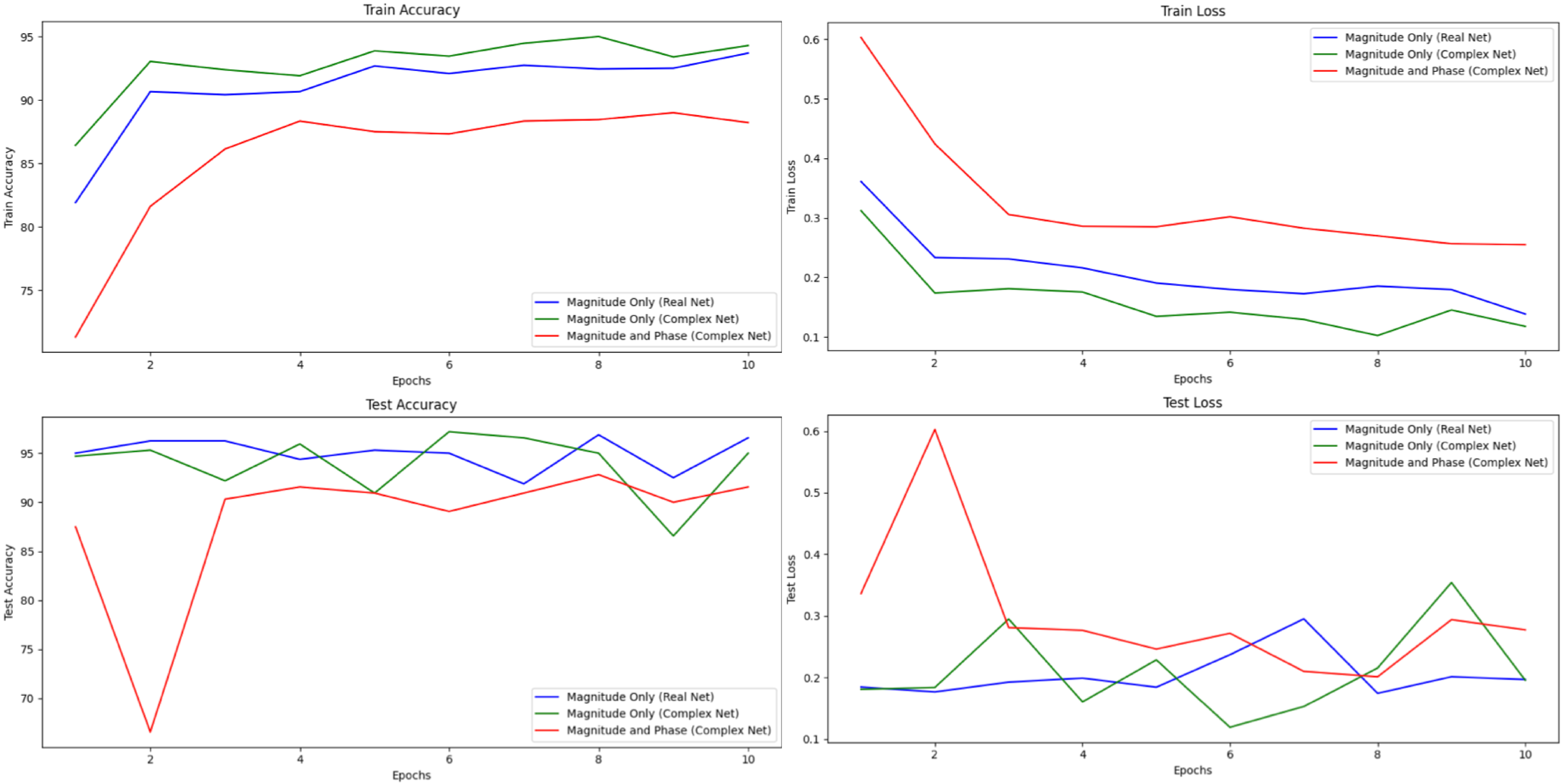
Real Valued Coeffs

Complex Valued Coeffs

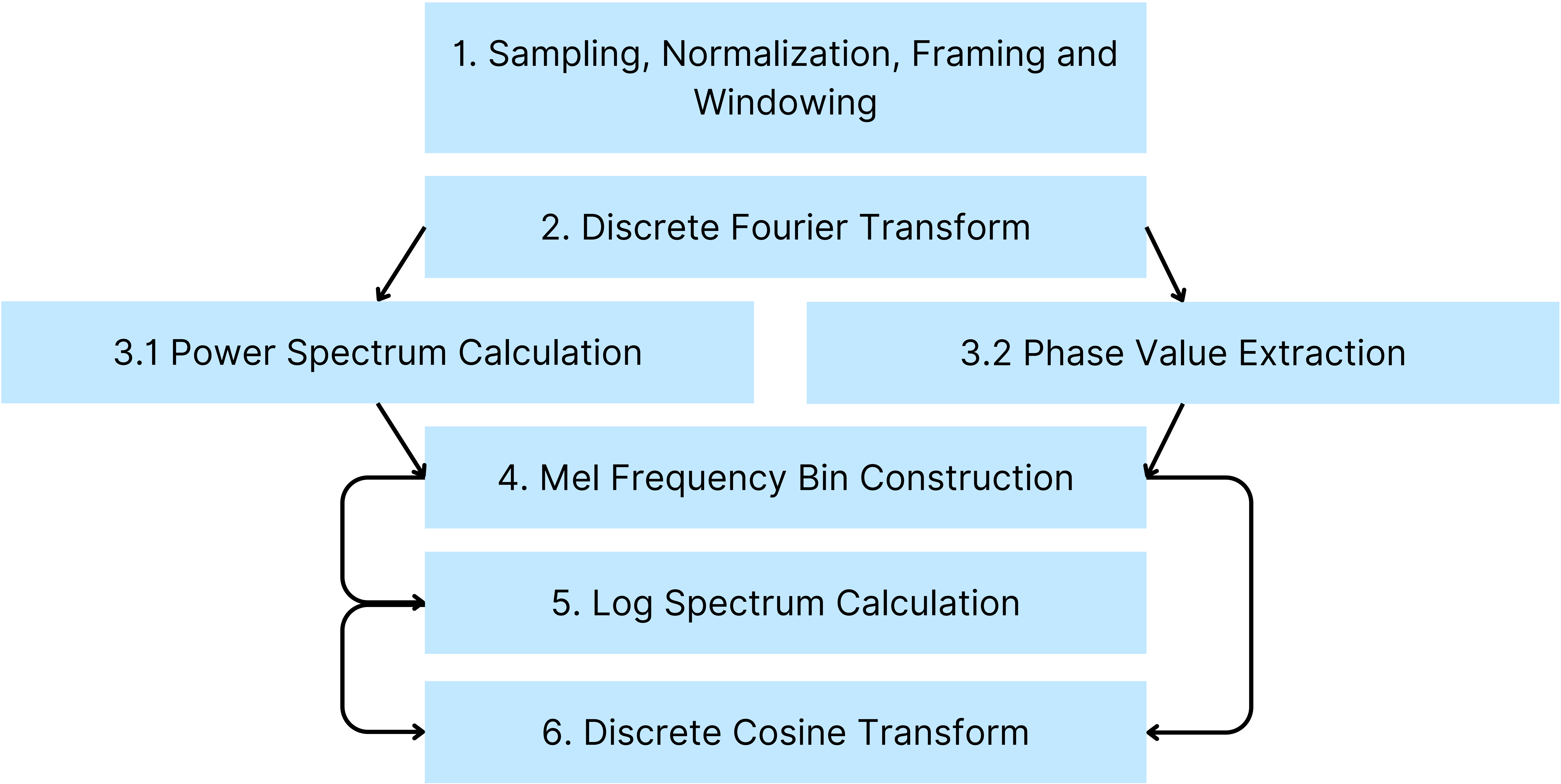
Part 1: All 10 Classes

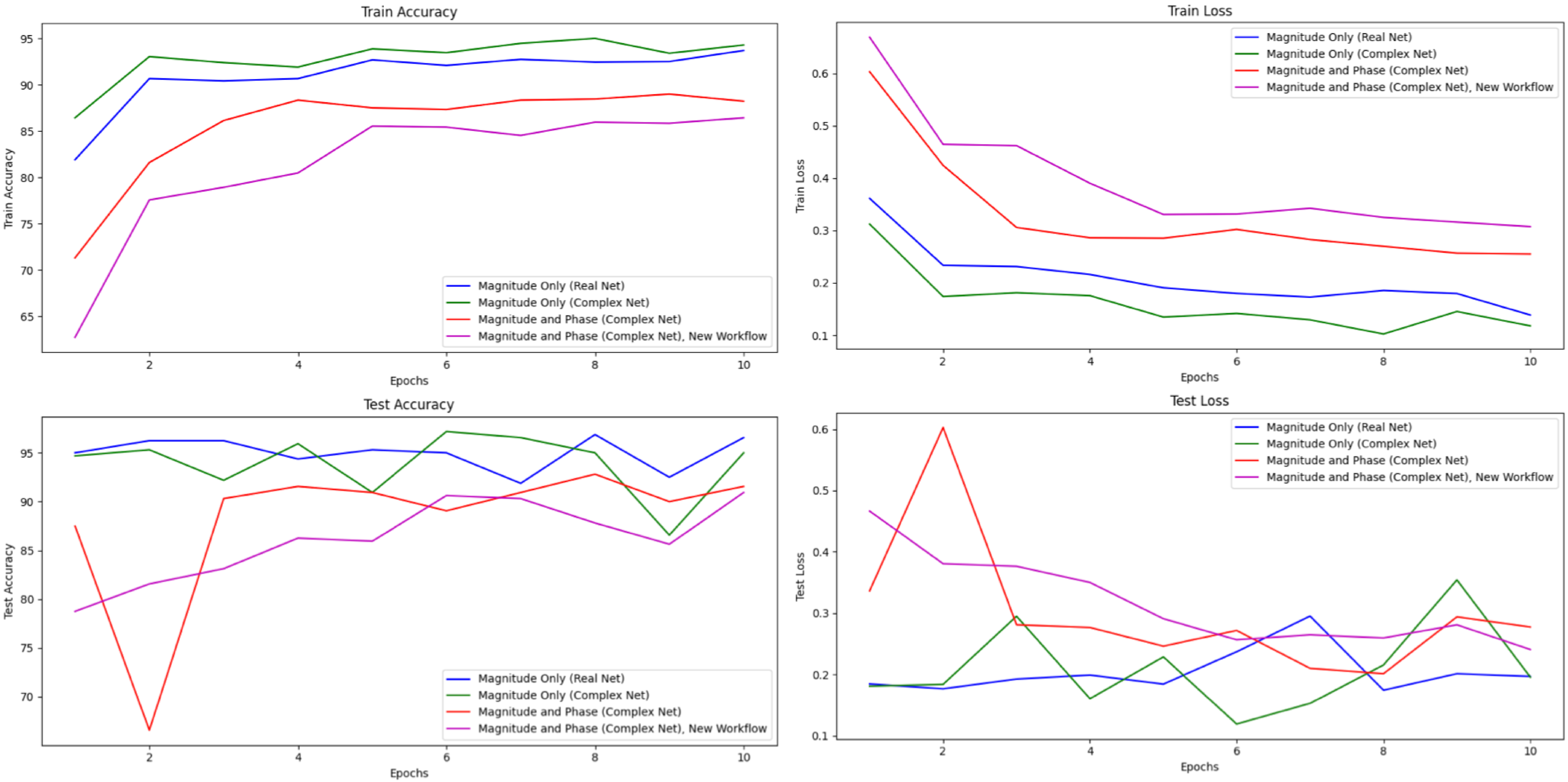


Part 2: Only 2 Classes



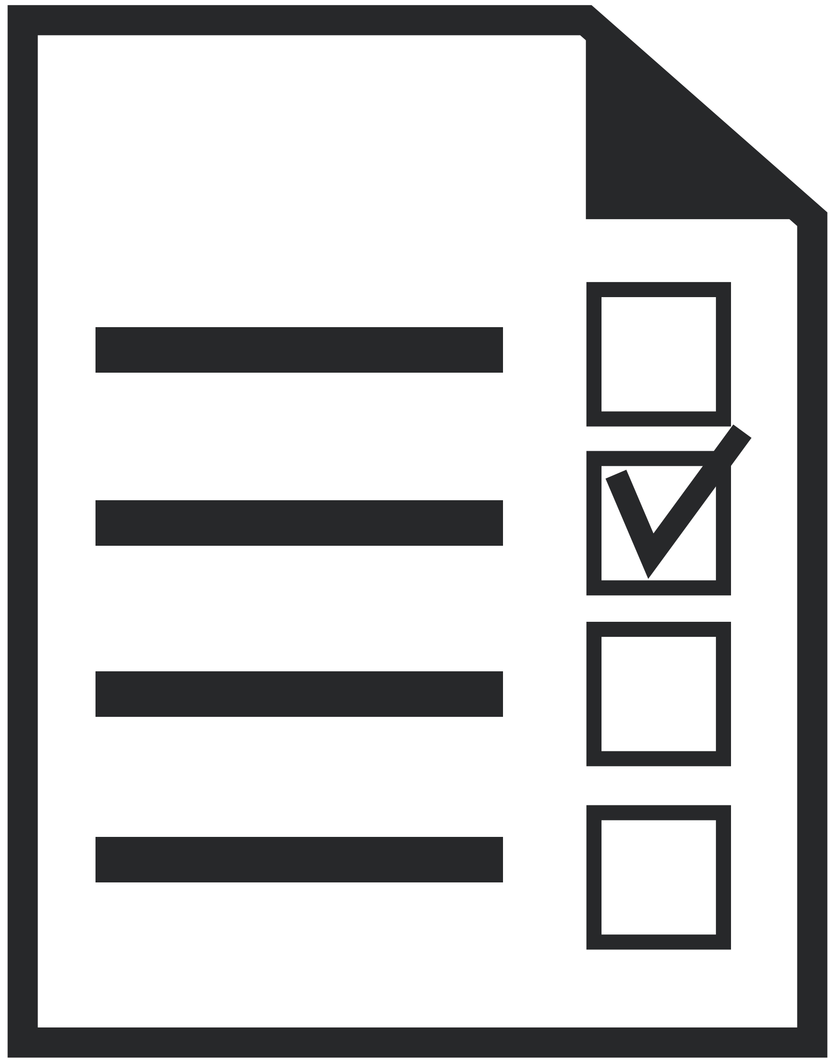
MFCC Extraction Workflow: II




Train Accuracy
Train Accuracy
Epochs
Magnitude Only (Real Net)
Magnitude Only (Complex Net)
Magnitude and Phase (Complex Net)
Magnitude and Phase (Complex Net), New Workflow
Train Loss
Train Loss
Epochs
Magnitude Only (Real Net)
Magnitude Only (Complex Net)
Magnitude and Phase (Complex Net)
Magnitude and Phase (Complex Net), New Workflow
Test Accuracy
Test Accuracy
Epochs
Magnitude Only (Real Net)
Magnitude Only (Complex Net)
Magnitude and Phase (Complex Net)
Magnitude and Phase (Complex Net), New Workflow
Test Loss
Test Loss
Epochs
Magnitude Only (Real Net)
Magnitude Only (Complex Net)
Magnitude and Phase (Complex Net)
Magnitude and Phase (Complex Net), New Workflow

Results 3

EXP3: Music Genre Dataset [Audio Signal Clasification]
Binary Variant Used
Implementation of Custom GNN Pipeline + ComplexNet
as from Previous Use Case



GNN Workflow: I

1. Real Valued MFCCs (From MFCC Extraction Workflow 2) [1, 13, 126]

2. GNN: Each node is a set of coeffs (Num Nodes: 13, each [126])

Path 1

Path 2

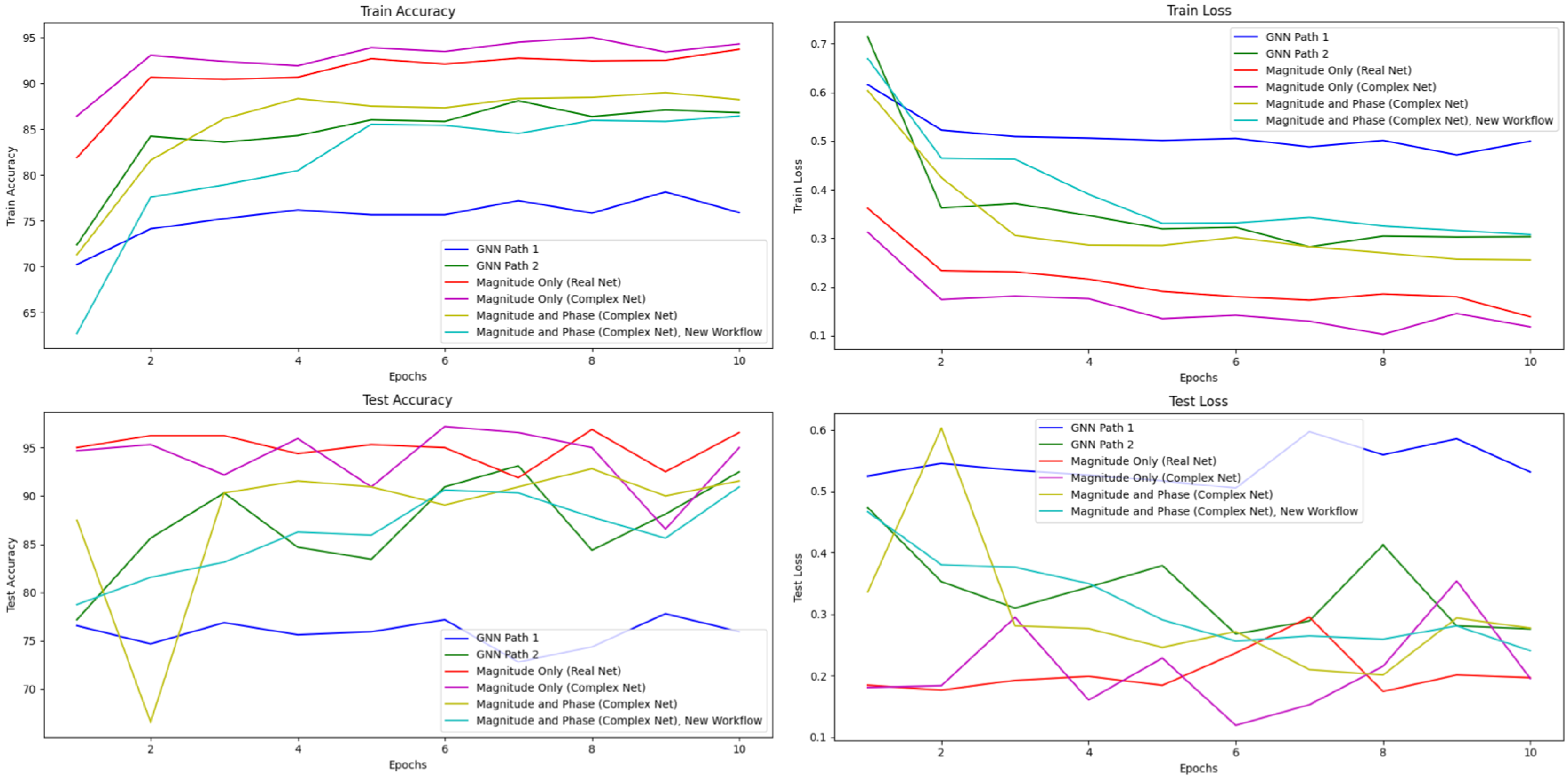
3.1 Constant Edge Weights (= 1.0)

3.2 Edge Weights: Mean phase difference

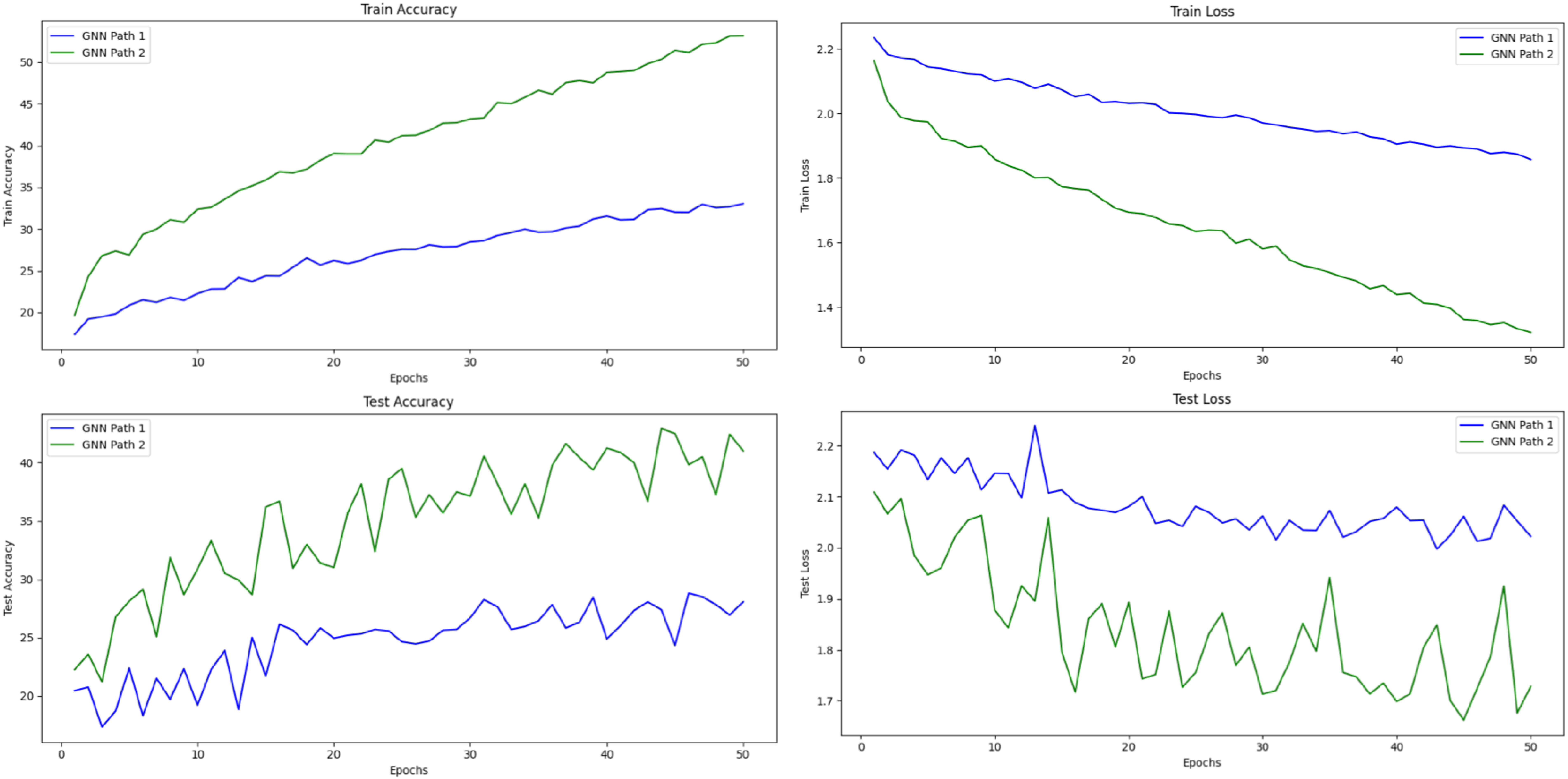
4. Absolute Values Extracted

5. ComplexNet (as from earlier)

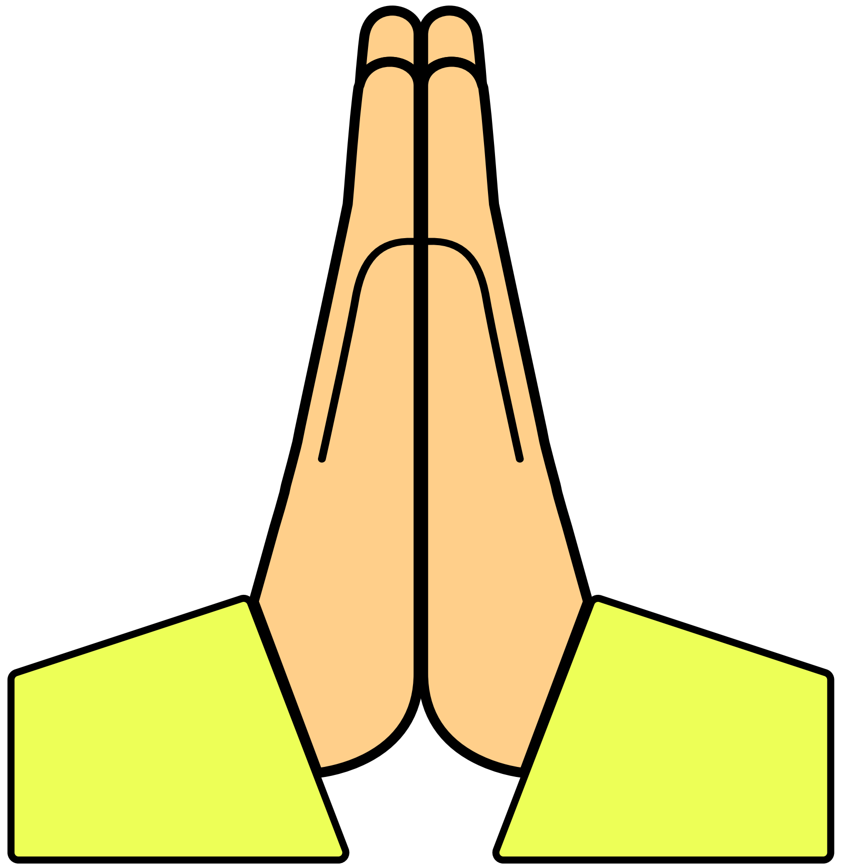
1. Edge Weight Effect for Binary



2. Edge Weight Effect for Multi



Thank You



Contact: Naman Agrawal @ +91 8850940094