Complex PyTorch for Music Genre Classification

In [28]:
```
# Complex pytorch
import torch
import torch.nn as nn
import torch.nn.functional as F
from torch.utils.data import DataLoader
from torchvision import datasets, transforms
from complexPyTorch.complexLayers import *
from complexPyTorch.complexFunctions import *
from torch_geometric.nn import GCNConv
from torch_geometric.data import Data

# Plot
import matplotlib.pyplot as plt
import seaborn as sns
import time

# Load Data
import numpy as np
import json
import os
import math
import librosa
import pathlib
from scipy.spatial.distance import cdist
from torch.utils.data import Dataset
from sklearn.model_selection import train_test_split
import random

# MFCCS
from scipy.io import wavfile
import scipy.fftpack as fft
from scipy.signal import get_window
```

Data Preparation

In [29]:
```
DATASET_PATH = "Data/train"
SAMPLE_RATE = 22050
TRACK_DURATION = 30 # measured in seconds
SAMPLES_PER_TRACK = SAMPLE_RATE * TRACK_DURATION
BATCH_SIZE = 32
NUM_EPOCHS = 50
```

In [30]:
```
genre_list = os.listdir(DATASET_PATH)
if '.DS_Store' in genre_list: genre_list.remove('.DS_Store')
genre_mappings = dict(zip(genre_list, range(len(genre_list))))
print(genre_mappings)
```

```
{'pop': 0, 'metal': 1, 'disco': 2, 'blues': 3, 'reggae': 4, 'classical':
5, 'rock': 6, 'hiphop': 7, 'country': 8, 'jazz': 9}
```

MFCCS

In [31]:
```
class MusicFeatureExtractorComplex2:
    def __init__(self, FFT_size=2048, HOP_SIZE=512, mel_filter_num=13, dc
        self.FFT_size = FFT_size
        self.HOP_SIZE = HOP_SIZE
        self.mel_filter_num = mel_filter_num
        self.dct_filter_num = dct_filter_num
        self.epsilon = 1e-10  # Added to log to avoid log10(0)

    def normalize_audio(self, audio):
        audio = audio / np.max(np.abs(audio))
        return audio

    def frame_audio(self, audio):
        frame_num = int((len(audio) - self.FFT_size) / self.HOP_SIZE) + 1
        frames = np.zeros((frame_num, self.FFT_size))
        for n in range(frame_num):
            frames[n] = audio[n * self.HOP_SIZE: n * self.HOP_SIZE + self
        return frames

    def freq_to_mel(self, freq):
        return 2595.0 * np.log10(1.0 + freq / 700.0)

    def met_to_freq(self, mels):
        return 700.0 * (10.0 ** (mels / 2595.0) - 1.0)

    def get_filter_points(self, fmin, fmax, sample_rate):
        fmin_mel = self.freq_to_mel(fmin)
        fmax_mel = self.freq_to_mel(fmax)
        mels = np.linspace(fmin_mel, fmax_mel, num=self.mel_filter_num +
        freqs = self.met_to_freq(mels)
        return np.floor((self.FFT_size + 1) / sample_rate * freqs).astype

    def get_filters(self, filter_points):
        filters = np.zeros((len(filter_points) - 2, int(self.FFT_size / 2
        for n in range(len(filter_points) - 2):
            filters[n, filter_points[n]: filter_points[n + 1]] = np.linsp
            filters[n, filter_points[n + 1]: filter_points[n + 2]] = np.l
        return filters

    def dct(self):
        basis = np.empty((self.dct_filter_num, self.mel_filter_num))
        basis[0, :] = 1.0 / np.sqrt(self.mel_filter_num)
        samples = np.arange(1, 2 * self.mel_filter_num, 2) * np.pi / (2.0
        for i in range(1, self.dct_filter_num):
            basis[i, :] = np.cos(i * samples) * np.sqrt(2.0 / self.mel_fi
        return basis

    def get_mfcc_features(self, audio, sample_rate):
        audio = self.normalize_audio(audio)
        audio_framed = self.frame_audio(audio)
        window = get_window("hann", self.FFT_size, fftbins=True)
        audio_win = audio_framed * window
        audio_winT = np.transpose(audio_win)
        audio_fft = np.empty((int(1 + self.FFT_size // 2), audio_winT.sha
        for n in range(audio_fft.shape[1]):
            audio_fft[:, n] = fft.fft(audio_winT[:, n], axis=0)[:audio_ff
        audio_fft = np.transpose(audio_fft)
        mag_fft = np.square(np.abs(audio_fft))
        phase_fft = np.angle(audio_fft)
```

```
        freq_min = 0
        freq_high = sample_rate / 2
        filter_points, mel_freqs = self.get_filter_points(freq_min, freq_
        filters = self.get_filters(filter_points)
        audio_filtered = np.dot(filters, np.transpose(mag_fft))
        phase_filtered = np.dot(filters, np.transpose(phase_fft))
        audio_filtered = np.maximum(audio_filtered, self.epsilon)  # Repl
        audio_log = 10.0 * np.log10(audio_filtered)
        dct_filters = self.dct()
        cepstral_coefficents = np.dot(dct_filters, audio_log)
        phase_coefficents = np.dot(dct_filters, phase_filtered)
        return np.array([cepstral_coefficents]), np.array([phase_coeffice
```

In [32]:

```
class GenreDatasetMFCC(Dataset):

    def __init__(self, train_path, n_fft=2048, hop_length=512, num_segmen
        cur_path = pathlib.Path(train_path)
        self.files = []
        for i in list(cur_path.rglob("*.wav")):
            for j in range(num_segments):
                self.files.append([j, i])
        self.samples_per_segment = int(SAMPLES_PER_TRACK / num_segments)
        self.n_fft = n_fft
        self.hop_length = hop_length
        self.num_segments = num_segments
        self.dct_filter_num = dct_filter_num
        self.training = training

    def apply_augmentations(self, signal):
        # Apply augmentations to the audio signal
        if random.random() < 0.5:
            signal = librosa.effects.pitch_shift(signal, sr=SAMPLE_RATE,
        if random.random() < 0.5:
            signal = librosa.effects.time_stretch(signal, rate=random.uni
        return signal

    def adjust_shape(self, sequence, max_sequence_length = 126):
        current_length = sequence.shape[2]
        if current_length < max_sequence_length:
            padding = np.zeros((1, 13, max_sequence_length - current_leng
            padded_sequence = np.concatenate((sequence, padding), axis=2)
        else:
            padded_sequence = sequence[:, :, :max_sequence_length]
        return padded_sequence

    def __len__(self):
        return len(self.files)

    def __getitem__(self, idx):
        cur_file = self.files[idx]
        d = cur_file[0]
        file_path = cur_file[1]
        target = genre_mappings[str(file_path).split("/")[2]]
        signal, sample_rate = librosa.load(file_path, sr=SAMPLE_RATE)
        start = self.samples_per_segment * d
        finish = start + self.samples_per_segment
        cur_signal = signal[start:finish]
        if self.training: cur_signal = self.apply_augmentations(cur_signa
        cur_mfcc = self.mfcc_extractor.get_mfcc_features(cur_signal, samp
        cur_mfcc = self.adjust_shape(cur_mfcc)
```

```
        return torch.tensor(cur_mfcc, dtype=torch.float32), target

class GenreDatasetPhaseMFCC2(GenreDatasetMFCC):

    def __init__(self, train_path, n_fft=2048, hop_length=512, num_segmen
        super().__init__(train_path, n_fft, hop_length, num_segments, mel
        self.mfcc_extractor = MusicFeatureExtractorComplex2(
            FFT_size=n_fft, HOP_SIZE=hop_length, mel_filter_num = mel_fil

    def __getitem__(self, idx):
        cur_file = self.files[idx]
        d = cur_file[0]
        file_path = cur_file[1]
        target = genre_mappings[str(file_path).split("/")[2]]
        signal, sample_rate = librosa.load(file_path, sr=SAMPLE_RATE)
        start = self.samples_per_segment * d
        finish = start + self.samples_per_segment
        cur_signal = signal[start:finish]
        if self.training: cur_signal = self.apply_augmentations(cur_signa
        cur_mfcc, cur_phase = self.mfcc_extractor.get_mfcc_features(cur_s
        cur_mfcc, cur_phase = self.adjust_shape(cur_mfcc), self.adjust_sh
        return torch.tensor(cur_mfcc, dtype=torch.float32), torch.tensor(
```

In [33]:

```
train_dataset = GenreDatasetPhaseMFCC2("Data/train/", n_fft=2048, hop_len
test_dataset = GenreDatasetPhaseMFCC2("Data/test/", n_fft=2048, hop_lengt
train_loader = torch.utils.data.DataLoader(dataset=train_dataset, shuffle
test_loader = torch.utils.data.DataLoader(dataset=test_dataset, shuffle=F
```

1. Simple Graph Net (Only magnitude)

In [34]:

```
def train(model, device, train_loader, test_loader, optimizer, epoch, met
    model.train()
    total_loss = 0
    correct = 0
    total_samples = len(train_loader.dataset)
    start_time = time.time()

    for batch_idx, (data, data2, target) in enumerate(train_loader):
        data, data2, target = data.to(device), data2.to(device), target.t
        optimizer.zero_grad()
        output = model(data)
        loss = F.nll_loss(output, target)
        loss.backward()
        optimizer.step()
        total_loss += loss.item()
        pred = output.argmax(dim=1, keepdim=True)
        correct += pred.eq(target.view_as(pred)).sum().item()

        if batch_idx % 100 == 0:
            batch_accuracy = 100. * correct / ((batch_idx + 1) * len(data
            print('Train Epoch: {:3} [{:6}/{:6} ({:3.0f}%)]\tLoss: {:.6f}
                epoch,
                batch_idx * len(data),
                total_samples,
                100. * batch_idx / len(train_loader),
                loss.item(),
                batch_accuracy)
            )
```

```
    end_time = time.time()
    epoch_times = metrics_dict['epoch_times']
    epoch_times.append(end_time - start_time)
    epoch_loss = total_loss / len(train_loader)
    epoch_accuracy = 100. * correct / total_samples
    train_losses = metrics_dict['train_losses']
    train_accuracies = metrics_dict['train_accuracies']
    train_losses.append(epoch_loss)
    train_accuracies.append(epoch_accuracy)
    print('Epoch {} - Time: {:.2f}s - Train Loss: {:.6f} - Train Accuracy

    # Evaluate on test data
    model.eval()
    test_loss = 0
    correct = 0
    with torch.no_grad():
        for data, data2, target in test_loader:
            data, data2, target = data.to(device), data2.to(device), targ
            output = model(data)
            test_loss += F.nll_loss(output, target, reduction='sum').item
            pred = output.argmax(dim=1, keepdim=True)
            correct += pred.eq(target.view_as(pred)).sum().item()

    test_loss /= len(test_loader.dataset)
    test_accuracy = 100. * correct / len(test_loader.dataset)
    test_losses = metrics_dict['test_losses']
    test_accuracies = metrics_dict['test_accuracies']
    test_losses.append(test_loss)
    test_accuracies.append(test_accuracy)
    print('Test Loss: {:.6f} - Test Accuracy: {:.2f}%\n'.format(test_loss
```

In [35]:

```
class ComplexGraphNet(nn.Module):
    def __init__(self):
        super(ComplexGraphNet, self).__init__()
        self.gnn_layer = GCNConv(in_channels=126, out_channels=126, node_
        self.conv1 = ComplexConv2d(1, 10, 2, 1)
        self.bn  = ComplexBatchNorm2d(10)
        self.conv2 = ComplexConv2d(10, 20, 2, 1)
        self.fc1 = ComplexLinear(30*2*20, 500)
        self.fc2 = ComplexLinear(500, 10)

    def forward(self, x):  # Pass edge_index for GNN
        batch_size, _, num_nodes, node_size = x.size()
        edge_index = torch.tensor([[i, j] for i in range(num_nodes) for j
        x = x.view(-1, num_nodes, node_size)  # Reshape for batch process
        x = self.gnn_layer(x, edge_index)
        x = x.unsqueeze(1)

        x = x.type(torch.complex64)
        x = self.conv1(x)
        x = complex_relu(x)
        x = complex_max_pool2d(x, 2, 2)
        x = self.bn(x)
        x = self.conv2(x)
        x = complex_relu(x)
        x = complex_max_pool2d(x, 2, 2)
        x = x.view(batch_size, -1)  # Reshape back to batched form
        x = self.fc1(x)
        x = complex_relu(x)
```

```
        x = self.fc2(x)
        x = x.abs()
        x = F.log_softmax(x, dim=1)
        return x

device = torch.device('cuda' if torch.cuda.is_available() else 'cpu')
model = ComplexGraphNet().to(device)
optimizer = torch.optim.SGD(model.parameters(), lr=0.01, momentum=0.9)

metrics_dict_e1 = {
    'epoch_times': [],
    'train_losses': [],
    'train_accuracies': [],
    'test_losses': [],
    'test_accuracies': []
}

for epoch in range(NUM_EPOCHS):
    train(model,
          device,
          train_loader,
          test_loader,
          optimizer,
          epoch,
          metrics_dict_e1)

print("-"*100)
print("-"*100)
print("FINAL RESULTS:")
print("-"*100)
for key, value in metrics_dict_e1.items():
    print(f'{key}: {value}')
```

```
Train Epoch:   0 [     0/  8390 (  0%)] Loss: 2.283815  Accuracy: 9.38%
Train Epoch:   0 [  3200/  8390 ( 38%)] Loss: 2.285956  Accuracy: 15.53%
Train Epoch:   0 [  6400/  8390 ( 76%)] Loss: 2.140933  Accuracy: 17.06%
Epoch 0 - Time: 233.97s - Train Loss: 2.234477 - Train Accuracy: 17.37%
Test Loss: 2.186637 - Test Accuracy: 20.44%

Train Epoch:   1 [     0/  8390 (  0%)] Loss: 2.173024  Accuracy: 25.00%
Train Epoch:   1 [  3200/  8390 ( 38%)] Loss: 2.406570  Accuracy: 19.15%
Train Epoch:   1 [  6400/  8390 ( 76%)] Loss: 2.102077  Accuracy: 19.45%
Epoch 1 - Time: 226.15s - Train Loss: 2.182805 - Train Accuracy: 19.19%
Test Loss: 2.154018 - Test Accuracy: 20.75%

Train Epoch:   2 [     0/  8390 (  0%)] Loss: 2.015391  Accuracy: 34.38%
Train Epoch:   2 [  3200/  8390 ( 38%)] Loss: 2.243402  Accuracy: 19.06%
Train Epoch:   2 [  6400/  8390 ( 76%)] Loss: 2.275811  Accuracy: 19.56%
Epoch 2 - Time: 236.52s - Train Loss: 2.171332 - Train Accuracy: 19.48%
Test Loss: 2.191294 - Test Accuracy: 17.31%

Train Epoch:   3 [     0/  8390 (  0%)] Loss: 2.239104  Accuracy: 9.38%
Train Epoch:   3 [  3200/  8390 ( 38%)] Loss: 2.304262  Accuracy: 20.39%
Train Epoch:   3 [  6400/  8390 ( 76%)] Loss: 2.138627  Accuracy: 20.20%
Epoch 3 - Time: 238.93s - Train Loss: 2.166512 - Train Accuracy: 19.83%
Test Loss: 2.181513 - Test Accuracy: 18.69%

Train Epoch:   4 [     0/  8390 (  0%)] Loss: 2.182257  Accuracy: 15.62%
Train Epoch:   4 [  3200/  8390 ( 38%)] Loss: 2.212053  Accuracy: 20.45%
Train Epoch:   4 [  6400/  8390 ( 76%)] Loss: 2.122830  Accuracy: 21.21%
Epoch 4 - Time: 239.38s - Train Loss: 2.143740 - Train Accuracy: 20.86%
Test Loss: 2.133375 - Test Accuracy: 22.38%

Train Epoch:   5 [     0/  8390 (  0%)] Loss: 2.226711  Accuracy: 18.75%
Train Epoch:   5 [  3200/  8390 ( 38%)] Loss: 2.121426  Accuracy: 21.10%
Train Epoch:   5 [  6400/  8390 ( 76%)] Loss: 1.933081  Accuracy: 21.39%
Epoch 5 - Time: 236.68s - Train Loss: 2.138812 - Train Accuracy: 21.49%
Test Loss: 2.176407 - Test Accuracy: 18.31%

Train Epoch:   6 [     0/  8390 (  0%)] Loss: 2.114547  Accuracy: 12.50%
Train Epoch:   6 [  3200/  8390 ( 38%)] Loss: 2.301475  Accuracy: 22.03%
Train Epoch:   6 [  6400/  8390 ( 76%)] Loss: 2.247961  Accuracy: 21.08%
Epoch 6 - Time: 237.03s - Train Loss: 2.130672 - Train Accuracy: 21.20%
Test Loss: 2.145782 - Test Accuracy: 21.50%

Train Epoch:   7 [     0/  8390 (  0%)] Loss: 2.018586  Accuracy: 37.50%
Train Epoch:   7 [  3200/  8390 ( 38%)] Loss: 2.011729  Accuracy: 21.04%
Train Epoch:   7 [  6400/  8390 ( 76%)] Loss: 2.357390  Accuracy: 22.05%
Epoch 7 - Time: 239.32s - Train Loss: 2.122056 - Train Accuracy: 21.80%
Test Loss: 2.176117 - Test Accuracy: 19.69%

Train Epoch:   8 [     0/  8390 (  0%)] Loss: 2.233269  Accuracy: 18.75%
Train Epoch:   8 [  3200/  8390 ( 38%)] Loss: 2.092553  Accuracy: 20.39%
Train Epoch:   8 [  6400/  8390 ( 76%)] Loss: 2.221566  Accuracy: 21.10%
Epoch 8 - Time: 242.28s - Train Loss: 2.119308 - Train Accuracy: 21.44%
Test Loss: 2.113507 - Test Accuracy: 22.31%

Train Epoch:   9 [     0/  8390 (  0%)] Loss: 1.932667  Accuracy: 37.50%
Train Epoch:   9 [  3200/  8390 ( 38%)] Loss: 2.014356  Accuracy: 22.71%
Train Epoch:   9 [  6400/  8390 ( 76%)] Loss: 2.107163  Accuracy: 22.62%
Epoch 9 - Time: 240.29s - Train Loss: 2.099407 - Train Accuracy: 22.24%
Test Loss: 2.145893 - Test Accuracy: 19.19%
```

```
Train Epoch:  10 [     0/  8390 (  0%)] Loss: 2.220515  Accuracy: 15.62%
Train Epoch:  10 [  3200/  8390 ( 38%)] Loss: 2.189619  Accuracy: 21.94%
Train Epoch:  10 [  6400/  8390 ( 76%)] Loss: 1.973737  Accuracy: 22.68%
Epoch 10 - Time: 241.49s - Train Loss: 2.108154 - Train Accuracy: 22.80%
Test Loss: 2.145267 - Test Accuracy: 22.25%

Train Epoch:  11 [     0/  8390 (  0%)] Loss: 2.138165  Accuracy: 18.75%
Train Epoch:  11 [  3200/  8390 ( 38%)] Loss: 2.291169  Accuracy: 23.05%
Train Epoch:  11 [  6400/  8390 ( 76%)] Loss: 2.173194  Accuracy: 22.84%
Epoch 11 - Time: 237.19s - Train Loss: 2.095983 - Train Accuracy: 22.82%
Test Loss: 2.097750 - Test Accuracy: 23.88%

Train Epoch:  12 [     0/  8390 (  0%)] Loss: 1.984829  Accuracy: 25.00%
Train Epoch:  12 [  3200/  8390 ( 38%)] Loss: 1.933785  Accuracy: 23.58%
Train Epoch:  12 [  6400/  8390 ( 76%)] Loss: 1.954490  Accuracy: 23.97%
Epoch 12 - Time: 233.34s - Train Loss: 2.078130 - Train Accuracy: 24.18%
Test Loss: 2.239533 - Test Accuracy: 18.81%

Train Epoch:  13 [     0/  8390 (  0%)] Loss: 2.180044  Accuracy: 15.62%
Train Epoch:  13 [  3200/  8390 ( 38%)] Loss: 1.909034  Accuracy: 24.29%
Train Epoch:  13 [  6400/  8390 ( 76%)] Loss: 2.191316  Accuracy: 23.17%
Epoch 13 - Time: 231.77s - Train Loss: 2.090801 - Train Accuracy: 23.71%
Test Loss: 2.107307 - Test Accuracy: 25.00%

Train Epoch:  14 [     0/  8390 (  0%)] Loss: 2.004154  Accuracy: 21.88%
Train Epoch:  14 [  3200/  8390 ( 38%)] Loss: 1.827421  Accuracy: 24.16%
Train Epoch:  14 [  6400/  8390 ( 76%)] Loss: 2.039760  Accuracy: 24.61%
Epoch 14 - Time: 228.70s - Train Loss: 2.073023 - Train Accuracy: 24.37%
Test Loss: 2.113122 - Test Accuracy: 21.69%

Train Epoch:  15 [     0/  8390 (  0%)] Loss: 2.098036  Accuracy: 12.50%
Train Epoch:  15 [  3200/  8390 ( 38%)] Loss: 1.838435  Accuracy: 23.73%
Train Epoch:  15 [  6400/  8390 ( 76%)] Loss: 2.227892  Accuracy: 24.07%
Epoch 15 - Time: 228.27s - Train Loss: 2.051294 - Train Accuracy: 24.35%
Test Loss: 2.088547 - Test Accuracy: 26.12%

Train Epoch:  16 [     0/  8390 (  0%)] Loss: 2.222553  Accuracy: 18.75%
Train Epoch:  16 [  3200/  8390 ( 38%)] Loss: 2.005491  Accuracy: 25.31%
Train Epoch:  16 [  6400/  8390 ( 76%)] Loss: 2.025543  Accuracy: 25.36%
Epoch 16 - Time: 228.22s - Train Loss: 2.059692 - Train Accuracy: 25.40%
Test Loss: 2.077507 - Test Accuracy: 25.62%

Train Epoch:  17 [     0/  8390 (  0%)] Loss: 2.034314  Accuracy: 18.75%
Train Epoch:  17 [  3200/  8390 ( 38%)] Loss: 1.842534  Accuracy: 28.56%
Train Epoch:  17 [  6400/  8390 ( 76%)] Loss: 2.120786  Accuracy: 27.05%
Epoch 17 - Time: 226.93s - Train Loss: 2.034155 - Train Accuracy: 26.52%
Test Loss: 2.073564 - Test Accuracy: 24.38%

Train Epoch:  18 [     0/  8390 (  0%)] Loss: 2.023822  Accuracy: 25.00%
Train Epoch:  18 [  3200/  8390 ( 38%)] Loss: 1.873845  Accuracy: 26.64%
Train Epoch:  18 [  6400/  8390 ( 76%)] Loss: 2.002745  Accuracy: 26.00%
Epoch 18 - Time: 227.17s - Train Loss: 2.036771 - Train Accuracy: 25.69%
Test Loss: 2.069038 - Test Accuracy: 25.81%

Train Epoch:  19 [     0/  8390 (  0%)] Loss: 2.088804  Accuracy: 21.88%
Train Epoch:  19 [  3200/  8390 ( 38%)] Loss: 1.972812  Accuracy: 25.93%
Train Epoch:  19 [  6400/  8390 ( 76%)] Loss: 1.902578  Accuracy: 25.47%
Epoch 19 - Time: 228.38s - Train Loss: 2.030699 - Train Accuracy: 26.22%
Test Loss: 2.080796 - Test Accuracy: 24.94%
```

```
Train Epoch:  20 [     0/  8390 (  0%)] Loss: 1.757594  Accuracy: 31.25%
Train Epoch:  20 [  3200/  8390 ( 38%)] Loss: 2.131952  Accuracy: 25.93%
Train Epoch:  20 [  6400/  8390 ( 76%)] Loss: 1.976331  Accuracy: 25.65%
Epoch 20 - Time: 234.30s - Train Loss: 2.032363 - Train Accuracy: 25.85%
Test Loss: 2.099731 - Test Accuracy: 25.19%

Train Epoch:  21 [     0/  8390 (  0%)] Loss: 1.994877  Accuracy: 21.88%
Train Epoch:  21 [  3200/  8390 ( 38%)] Loss: 2.212613  Accuracy: 25.96%
Train Epoch:  21 [  6400/  8390 ( 76%)] Loss: 1.958855  Accuracy: 25.64%
Epoch 21 - Time: 225.82s - Train Loss: 2.027406 - Train Accuracy: 26.23%
Test Loss: 2.047946 - Test Accuracy: 25.31%

Train Epoch:  22 [     0/  8390 (  0%)] Loss: 2.057194  Accuracy: 28.12%
Train Epoch:  22 [  3200/  8390 ( 38%)] Loss: 1.830850  Accuracy: 27.51%
Train Epoch:  22 [  6400/  8390 ( 76%)] Loss: 1.977901  Accuracy: 27.15%
Epoch 22 - Time: 225.02s - Train Loss: 2.001441 - Train Accuracy: 26.96%
Test Loss: 2.053808 - Test Accuracy: 25.69%

Train Epoch:  23 [     0/  8390 (  0%)] Loss: 2.087957  Accuracy: 18.75%
Train Epoch:  23 [  3200/  8390 ( 38%)] Loss: 1.934952  Accuracy: 28.22%
Train Epoch:  23 [  6400/  8390 ( 76%)] Loss: 2.093279  Accuracy: 27.61%
Epoch 23 - Time: 224.18s - Train Loss: 1.999862 - Train Accuracy: 27.32%
Test Loss: 2.041623 - Test Accuracy: 25.56%

Train Epoch:  24 [     0/  8390 (  0%)] Loss: 1.804892  Accuracy: 37.50%
Train Epoch:  24 [  3200/  8390 ( 38%)] Loss: 2.206030  Accuracy: 27.78%
Train Epoch:  24 [  6400/  8390 ( 76%)] Loss: 1.932141  Accuracy: 27.08%
Epoch 24 - Time: 224.50s - Train Loss: 1.996928 - Train Accuracy: 27.57%
Test Loss: 2.081249 - Test Accuracy: 24.62%

Train Epoch:  25 [     0/  8390 (  0%)] Loss: 1.818998  Accuracy: 37.50%
Train Epoch:  25 [  3200/  8390 ( 38%)] Loss: 2.048907  Accuracy: 28.50%
Train Epoch:  25 [  6400/  8390 ( 76%)] Loss: 2.009765  Accuracy: 27.57%
Epoch 25 - Time: 224.85s - Train Loss: 1.990574 - Train Accuracy: 27.56%
Test Loss: 2.069043 - Test Accuracy: 24.44%

Train Epoch:  26 [     0/  8390 (  0%)] Loss: 1.843599  Accuracy: 37.50%
Train Epoch:  26 [  3200/  8390 ( 38%)] Loss: 2.136900  Accuracy: 27.44%
Train Epoch:  26 [  6400/  8390 ( 76%)] Loss: 1.867156  Accuracy: 27.89%
Epoch 26 - Time: 227.72s - Train Loss: 1.986620 - Train Accuracy: 28.12%
Test Loss: 2.048724 - Test Accuracy: 24.69%

Train Epoch:  27 [     0/  8390 (  0%)] Loss: 1.889203  Accuracy: 31.25%
Train Epoch:  27 [  3200/  8390 ( 38%)] Loss: 1.977899  Accuracy: 28.34%
Train Epoch:  27 [  6400/  8390 ( 76%)] Loss: 2.176792  Accuracy: 27.83%
Epoch 27 - Time: 226.10s - Train Loss: 1.995100 - Train Accuracy: 27.88%
Test Loss: 2.056802 - Test Accuracy: 25.62%

Train Epoch:  28 [     0/  8390 (  0%)] Loss: 1.990392  Accuracy: 31.25%
Train Epoch:  28 [  3200/  8390 ( 38%)] Loss: 1.947343  Accuracy: 27.88%
Train Epoch:  28 [  6400/  8390 ( 76%)] Loss: 2.123044  Accuracy: 28.00%
Epoch 28 - Time: 223.05s - Train Loss: 1.985914 - Train Accuracy: 27.91%
Test Loss: 2.034711 - Test Accuracy: 25.69%

Train Epoch:  29 [     0/  8390 (  0%)] Loss: 1.824628  Accuracy: 43.75%
Train Epoch:  29 [  3200/  8390 ( 38%)] Loss: 1.940703  Accuracy: 29.21%
Train Epoch:  29 [  6400/  8390 ( 76%)] Loss: 2.051470  Accuracy: 28.28%
Epoch 29 - Time: 223.30s - Train Loss: 1.970584 - Train Accuracy: 28.46%
Test Loss: 2.062386 - Test Accuracy: 26.69%
```

```
Train Epoch:  30 [     0/  8390 (  0%)] Loss: 1.932767  Accuracy: 31.25%
Train Epoch:  30 [  3200/  8390 ( 38%)] Loss: 2.009117  Accuracy: 29.02%
Train Epoch:  30 [  6400/  8390 ( 76%)] Loss: 1.938020  Accuracy: 28.51%
Epoch 30 - Time: 221.56s - Train Loss: 1.964117 - Train Accuracy: 28.62%
Test Loss: 2.015479 - Test Accuracy: 28.25%

Train Epoch:  31 [     0/  8390 (  0%)] Loss: 1.612053  Accuracy: 43.75%
Train Epoch:  31 [  3200/  8390 ( 38%)] Loss: 1.733670  Accuracy: 29.73%
Train Epoch:  31 [  6400/  8390 ( 76%)] Loss: 1.920024  Accuracy: 29.20%
Epoch 31 - Time: 223.11s - Train Loss: 1.956306 - Train Accuracy: 29.24%
Test Loss: 2.053814 - Test Accuracy: 27.62%

Train Epoch:  32 [     0/  8390 (  0%)] Loss: 1.901363  Accuracy: 25.00%
Train Epoch:  32 [  3200/  8390 ( 38%)] Loss: 1.922382  Accuracy: 30.14%
Train Epoch:  32 [  6400/  8390 ( 76%)] Loss: 2.035112  Accuracy: 29.73%
Epoch 32 - Time: 220.86s - Train Loss: 1.951182 - Train Accuracy: 29.58%
Test Loss: 2.034440 - Test Accuracy: 25.69%

Train Epoch:  33 [     0/  8390 (  0%)] Loss: 1.991580  Accuracy: 25.00%
Train Epoch:  33 [  3200/  8390 ( 38%)] Loss: 1.911169  Accuracy: 30.82%
Train Epoch:  33 [  6400/  8390 ( 76%)] Loss: 1.893150  Accuracy: 30.53%
Epoch 33 - Time: 225.68s - Train Loss: 1.944604 - Train Accuracy: 29.99%
Test Loss: 2.033751 - Test Accuracy: 25.94%

Train Epoch:  34 [     0/  8390 (  0%)] Loss: 1.752453  Accuracy: 31.25%
Train Epoch:  34 [  3200/  8390 ( 38%)] Loss: 1.883572  Accuracy: 30.17%
Train Epoch:  34 [  6400/  8390 ( 76%)] Loss: 1.837702  Accuracy: 29.49%
Epoch 34 - Time: 223.98s - Train Loss: 1.946527 - Train Accuracy: 29.62%
Test Loss: 2.073126 - Test Accuracy: 26.44%

Train Epoch:  35 [     0/  8390 (  0%)] Loss: 2.025464  Accuracy: 25.00%
Train Epoch:  35 [  3200/  8390 ( 38%)] Loss: 1.954219  Accuracy: 29.95%
Train Epoch:  35 [  6400/  8390 ( 76%)] Loss: 1.886833  Accuracy: 29.94%
Epoch 35 - Time: 223.24s - Train Loss: 1.936648 - Train Accuracy: 29.68%
Test Loss: 2.020539 - Test Accuracy: 27.81%

Train Epoch:  36 [     0/  8390 (  0%)] Loss: 1.900638  Accuracy: 25.00%
Train Epoch:  36 [  3200/  8390 ( 38%)] Loss: 1.862369  Accuracy: 30.14%
Train Epoch:  36 [  6400/  8390 ( 76%)] Loss: 1.809317  Accuracy: 30.36%
Epoch 36 - Time: 227.75s - Train Loss: 1.942515 - Train Accuracy: 30.13%
Test Loss: 2.031603 - Test Accuracy: 25.81%

Train Epoch:  37 [     0/  8390 (  0%)] Loss: 2.025652  Accuracy: 21.88%
Train Epoch:  37 [  3200/  8390 ( 38%)] Loss: 1.811957  Accuracy: 30.32%
Train Epoch:  37 [  6400/  8390 ( 76%)] Loss: 2.013695  Accuracy: 30.66%
Epoch 37 - Time: 222.05s - Train Loss: 1.927433 - Train Accuracy: 30.36%
Test Loss: 2.051604 - Test Accuracy: 26.31%

Train Epoch:  38 [     0/  8390 (  0%)] Loss: 1.649459  Accuracy: 43.75%
Train Epoch:  38 [  3200/  8390 ( 38%)] Loss: 2.124419  Accuracy: 31.37%
Train Epoch:  38 [  6400/  8390 ( 76%)] Loss: 1.922676  Accuracy: 30.77%
Epoch 38 - Time: 230.87s - Train Loss: 1.921546 - Train Accuracy: 31.20%
Test Loss: 2.057478 - Test Accuracy: 28.44%

Train Epoch:  39 [     0/  8390 (  0%)] Loss: 2.099957  Accuracy: 18.75%
Train Epoch:  39 [  3200/  8390 ( 38%)] Loss: 2.118045  Accuracy: 31.87%
Train Epoch:  39 [  6400/  8390 ( 76%)] Loss: 1.844247  Accuracy: 31.67%
Epoch 39 - Time: 227.03s - Train Loss: 1.904664 - Train Accuracy: 31.55%
Test Loss: 2.079740 - Test Accuracy: 24.88%
```

```
Train Epoch:  40 [     0/  8390 (  0%)] Loss: 2.131852  Accuracy: 25.00%
Train Epoch:  40 [  3200/  8390 ( 38%)] Loss: 1.620973  Accuracy: 31.68%
Train Epoch:  40 [  6400/  8390 ( 76%)] Loss: 2.101529  Accuracy: 31.31%
Epoch 40 - Time: 226.69s - Train Loss: 1.911417 - Train Accuracy: 31.11%
Test Loss: 2.053170 - Test Accuracy: 26.00%

Train Epoch:  41 [     0/  8390 (  0%)] Loss: 1.750316  Accuracy: 34.38%
Train Epoch:  41 [  3200/  8390 ( 38%)] Loss: 1.904465  Accuracy: 30.69%
Train Epoch:  41 [  6400/  8390 ( 76%)] Loss: 1.963924  Accuracy: 31.19%
Epoch 41 - Time: 225.56s - Train Loss: 1.904321 - Train Accuracy: 31.17%
Test Loss: 2.053985 - Test Accuracy: 27.31%

Train Epoch:  42 [     0/  8390 (  0%)] Loss: 1.974954  Accuracy: 28.12%
Train Epoch:  42 [  3200/  8390 ( 38%)] Loss: 1.891893  Accuracy: 31.96%
Train Epoch:  42 [  6400/  8390 ( 76%)] Loss: 1.921841  Accuracy: 32.59%
Epoch 42 - Time: 225.97s - Train Loss: 1.895247 - Train Accuracy: 32.32%
Test Loss: 1.997389 - Test Accuracy: 28.06%

Train Epoch:  43 [     0/  8390 (  0%)] Loss: 1.914194  Accuracy: 31.25%
Train Epoch:  43 [  3200/  8390 ( 38%)] Loss: 1.982564  Accuracy: 32.67%
Train Epoch:  43 [  6400/  8390 ( 76%)] Loss: 2.258419  Accuracy: 32.65%
Epoch 43 - Time: 223.78s - Train Loss: 1.899233 - Train Accuracy: 32.44%
Test Loss: 2.024002 - Test Accuracy: 27.38%

Train Epoch:  44 [     0/  8390 (  0%)] Loss: 1.560061  Accuracy: 50.00%
Train Epoch:  44 [  3200/  8390 ( 38%)] Loss: 1.845327  Accuracy: 32.92%
Train Epoch:  44 [  6400/  8390 ( 76%)] Loss: 2.210847  Accuracy: 32.40%
Epoch 44 - Time: 226.01s - Train Loss: 1.893218 - Train Accuracy: 32.04%
Test Loss: 2.061902 - Test Accuracy: 24.31%

Train Epoch:  45 [     0/  8390 (  0%)] Loss: 1.775600  Accuracy: 43.75%
Train Epoch:  45 [  3200/  8390 ( 38%)] Loss: 1.951963  Accuracy: 32.24%
Train Epoch:  45 [  6400/  8390 ( 76%)] Loss: 1.674304  Accuracy: 31.95%
Epoch 45 - Time: 225.07s - Train Loss: 1.889554 - Train Accuracy: 32.03%
Test Loss: 2.012643 - Test Accuracy: 28.81%

Train Epoch:  46 [     0/  8390 (  0%)] Loss: 1.857784  Accuracy: 34.38%
Train Epoch:  46 [  3200/  8390 ( 38%)] Loss: 1.889494  Accuracy: 33.79%
Train Epoch:  46 [  6400/  8390 ( 76%)] Loss: 1.861357  Accuracy: 32.76%
Epoch 46 - Time: 225.23s - Train Loss: 1.875798 - Train Accuracy: 32.97%
Test Loss: 2.018404 - Test Accuracy: 28.50%

Train Epoch:  47 [     0/  8390 (  0%)] Loss: 2.024217  Accuracy: 31.25%
Train Epoch:  47 [  3200/  8390 ( 38%)] Loss: 1.554365  Accuracy: 32.58%
Train Epoch:  47 [  6400/  8390 ( 76%)] Loss: 1.847797  Accuracy: 31.98%
Epoch 47 - Time: 224.48s - Train Loss: 1.879617 - Train Accuracy: 32.55%
Test Loss: 2.083142 - Test Accuracy: 27.81%

Train Epoch:  48 [     0/  8390 (  0%)] Loss: 1.864645  Accuracy: 37.50%
Train Epoch:  48 [  3200/  8390 ( 38%)] Loss: 2.004141  Accuracy: 32.12%
Train Epoch:  48 [  6400/  8390 ( 76%)] Loss: 1.954044  Accuracy: 32.56%
Epoch 48 - Time: 222.04s - Train Loss: 1.873967 - Train Accuracy: 32.67%
Test Loss: 2.052414 - Test Accuracy: 26.94%

Train Epoch:  49 [     0/  8390 (  0%)] Loss: 1.916777  Accuracy: 40.62%
Train Epoch:  49 [  3200/  8390 ( 38%)] Loss: 1.728014  Accuracy: 33.48%
Train Epoch:  49 [  6400/  8390 ( 76%)] Loss: 1.615704  Accuracy: 33.26%
Epoch 49 - Time: 1162.20s - Train Loss: 1.856905 - Train Accuracy: 33.05%
Test Loss: 2.021979 - Test Accuracy: 28.06%
```

```
--------------------------------------------------------------------------------------------
--------------------------------------------------------------------------------------------
FINAL RESULTS:
--------------------------------------------------------------------------------------------
epoch_times: [233.97046875953674, 226.14852595329285, 236.52325081825256,
238.92535185813904, 239.38257002830505, 236.6788511276245, 237.02662801742
554, 239.32301020622253, 242.27710700035095, 240.29133009910583, 241.49365
496635437, 237.19384503364563, 233.34383893013, 231.7748110294342, 228.695
59788703918, 228.27052807807922, 228.22335124015808, 226.9341676235199, 22
7.17398810386658, 228.38212084770203, 234.2978479862213, 225.821125984191
9, 225.02131295204163, 224.1751549243927, 224.49942111968994, 224.85096287
727356, 227.71697902679443, 226.1047580242157, 223.05186772346497, 223.297
59693145752, 221.56133818626404, 223.107741355896, 220.85710501670837, 22
5.68304800987244, 223.97671675682068, 223.2411026954651, 227.7453482151031
5, 222.0526521205902, 230.87250208854675, 227.02583384513855, 226.69209814
071655, 225.56455397605896, 225.97482013702393, 223.7847821712494, 226.007
67278671265, 225.06873416900635, 225.23424696922302, 224.47550320625305, 2
22.03592991828918, 1162.1967041492462]
train_losses: [2.234476645029228, 2.1828049061862567, 2.171332196879933,
2.166512229515396, 2.1437400169954954, 2.138811717051586, 2.13067216045073
86, 2.1220557052670546, 2.1193076758894303, 2.099406872996847, 2.108154222
2554446, 2.0959825861545007, 2.0781302060789733, 2.0908009400804533, 2.073
023048066001, 2.051294315862292, 2.0596924214872696, 2.0341545625497366,
2.0367706076789447, 2.0306985701313454, 2.0323631217461506, 2.027406216577
7194, 2.0014411761560513, 1.9998621972462602, 1.9969276835900227, 1.990574
1105552848, 1.9866200831100231, 1.9950999772275677, 1.985913817664139, 1.9
705841104492887, 1.9641166929980272, 1.9563061344714565, 1.951182177958597
6, 1.9446035810099302, 1.9465266179492455, 1.9366483128707828, 1.942514810
853332, 1.9274329051716637, 1.9215455496584186, 1.9046635550397042, 1.9114
173896440112, 1.9043212164449328, 1.8952473711421471, 1.899233222917746,
1.8932179776766829, 1.8895541998266263, 1.8757984788363216, 1.879616890699
8088, 1.873966821732412, 1.8569049575856624]
train_accuracies: [17.36591179976162, 19.189511323003575, 19.4755661501787
86, 19.833134684147794, 20.858164481525627, 21.489868891537544, 21.2038140
64362337, 21.799761620977353, 21.442193087008345, 22.240762812872468, 22.8
00953516090583, 22.824791418355186, 24.183551847437425, 23.70679380214541
3, 24.37425506555423, 24.35041716328963, 25.399284862932063, 26.5196662693
68294, 25.68533969010727, 26.221692491060786, 25.852205005959476, 26.23361
1442193087, 26.96066746126341, 27.31823599523242, 27.568533969010726, 27.5
5661501787843, 28.116805721096544, 27.878426698450536, 27.914183551847437,
28.462455303933254, 28.61740166865316, 29.237187127532778, 29.582836710369
488, 29.9880810488677, 29.61859356376639, 29.67818831942789, 30.1311084624
55305, 30.357568533969012, 31.203814064362337, 31.549463647199048, 31.1084
62455303933, 31.168057210965436, 32.32419547079857, 32.44338498212157, 32.
038140643623365, 32.02622169249106, 32.967818831942786, 32.55065554231228,
32.66984505363528, 33.05125148986889]
test_losses: [2.1866367852687834, 2.154017615318298, 2.1912943744659423,
2.1815134119987487, 2.1333746206760407, 2.176407445669174, 2.1457816970348
36, 2.176116580963135, 2.1135067558288574, 2.145893280506134, 2.1452674436
569215, 2.0977504682540893, 2.2395331597328187, 2.1073072028160094, 2.1131
2224149704, 2.088547270298004, 2.077507041692734, 2.0735635244846344, 2.06
90382504463196, 2.080795907974243, 2.0997307825088503, 2.0479457080364227,
2.0538084936141967, 2.041623057126999, 2.0812490618228914, 2.0690434098243
715, 2.0487236738204957, 2.056801930665969 7, 2.0347112727165224, 2.0623856
341838835, 2.015479021072388, 2.0538138830661774, 2.0344402396678927, 2.03
37510764598847, 2.073126270771026 7, 2.0205389261245728, 2.031603275537490
6, 2.0516035866737368, 2.057477984428406, 2.079739997386932, 2.05316980123
```

```
5199, 2.0539849162101746, 1.9973892450332642, 2.0240018010139464, 2.061902
1534919737, 2.0126427090168, 2.018404322862625, 2.0831423473358153, 2.0524
144637584687, 2.02197856426239]
test_accuracies: [20.4375, 20.75, 17.3125, 18.6875, 22.375, 18.3125, 21.5,
19.6875, 22.3125, 19.1875, 22.25, 23.875, 18.8125, 25.0, 21.6875, 26.125,
25.625, 24.375, 25.8125, 24.9375, 25.1875, 25.3125, 25.6875, 25.5625, 24.6
25, 24.4375, 24.6875, 25.625, 25.6875, 26.6875, 28.25, 27.625, 25.6875, 2
5.9375, 26.4375, 27.8125, 25.8125, 26.3125, 28.4375, 24.875, 26.0, 27.312
5, 28.0625, 27.375, 24.3125, 28.8125, 28.5, 27.8125, 26.9375, 28.0625]
```

2. Simple Graph Net (Magniude + phase wieghts)

In [36]:
```
def train(model, device, train_loader, test_loader, optimizer, epoch, met
    model.train()
    total_loss = 0
    correct = 0
    total_samples = len(train_loader.dataset)
    start_time = time.time()

    for batch_idx, (data, data2, target) in enumerate(train_loader):
        data, data2, target = data.to(device), data2.to(device), target.t
        optimizer.zero_grad()
        output = model([data, data2])
        loss = F.nll_loss(output, target)
        loss.backward()
        optimizer.step()
        total_loss += loss.item()
        pred = output.argmax(dim=1, keepdim=True)
        correct += pred.eq(target.view_as(pred)).sum().item()

        if batch_idx % 100 == 0:
            batch_accuracy = 100. * correct / ((batch_idx + 1) * len(data
            print('Train Epoch: {:3} [{:6}/{:6} ({:3.0f}%)]\tLoss: {:.6f}
                epoch,
                batch_idx * len(data),
                total_samples,
                100. * batch_idx / len(train_loader),
                loss.item(),
                batch_accuracy)
            )

    end_time = time.time()
    epoch_times = metrics_dict['epoch_times']
    epoch_times.append(end_time - start_time)
    epoch_loss = total_loss / len(train_loader)
    epoch_accuracy = 100. * correct / total_samples
    train_losses = metrics_dict['train_losses']
    train_accuracies = metrics_dict['train_accuracies']
    train_losses.append(epoch_loss)
    train_accuracies.append(epoch_accuracy)
    print('Epoch {} - Time: {:.2f}s - Train Loss: {:.6f} - Train Accuracy

    # Evaluate on test data
    model.eval()
    test_loss = 0
    correct = 0
    with torch.no_grad():
        for data, data2, target in test_loader:
```

```
            data, data2, target = data.to(device), data2.to(device), targ
            output = model([data, data2])
            test_loss += F.nll_loss(output, target, reduction='sum').item
            pred = output.argmax(dim=1, keepdim=True)
            correct += pred.eq(target.view_as(pred)).sum().item()

    test_loss /= len(test_loader.dataset)
    test_accuracy = 100. * correct / len(test_loader.dataset)
    test_losses = metrics_dict['test_losses']
    test_accuracies = metrics_dict['test_accuracies']
    test_losses.append(test_loss)
    test_accuracies.append(test_accuracy)
    print('Test Loss: {:.6f} - Test Accuracy: {:.2f}%\n'.format(test_loss
```

In [37]:

```
class ComplexGraphNet(nn.Module):
    def __init__(self):
        super(ComplexGraphNet, self).__init__()
        self.gnn_layer = GCNConv(in_channels=126, out_channels=126, node_
        self.conv1 = ComplexConv2d(1, 10, 2, 1)
        self.bn  = ComplexBatchNorm2d(10)
        self.conv2 = ComplexConv2d(10, 20, 2, 1)
        self.fc1 = ComplexLinear(30*2*20, 500)
        self.fc2 = ComplexLinear(500, 10)

    def forward(self, x):  # Pass edge_index for GNN
        x, phase_data = x[0], x[1]
        batch_size, _, num_nodes, node_size = x.size()
        edge_index = torch.tensor([[i, j] for i in range(num_nodes) for j
        phase_data = torch.mean(phase_data.view(-1, num_nodes, node_size)
        edge_weight = torch.tensor([torch.mean(np.abs(phase_data[edge_ind
                                                      phase_data[edge_ind
                                    for i in range(len(edge_index[0]))])
        x = x.view(-1, num_nodes, node_size)  # Reshape for batch process
        x = self.gnn_layer(x, edge_index, edge_weight)
        x = x.unsqueeze(1)

        x = x.type(torch.complex64)
        x = self.conv1(x)
        x = complex_relu(x)
        x = complex_max_pool2d(x, 2, 2)
        x = self.bn(x)
        x = self.conv2(x)
        x = complex_relu(x)
        x = complex_max_pool2d(x, 2, 2)
        x = x.view(batch_size, -1)  # Reshape back to batched form
        x = self.fc1(x)
        x = complex_relu(x)
        x = self.fc2(x)
        x = x.abs()
        x = F.log_softmax(x, dim=1)
        return x

device = torch.device('cuda' if torch.cuda.is_available() else 'cpu')
model = ComplexGraphNet().to(device)
optimizer = torch.optim.SGD(model.parameters(), lr=0.01, momentum=0.9)

metrics_dict_e2 = {
    'epoch_times': [],
    'train_losses': [],
    'train_accuracies': [],
```

```
    'test_losses': [],
    'test_accuracies': []
}

for epoch in range(NUM_EPOCHS):
    train(model,
          device,
          train_loader,
          test_loader,
          optimizer,
          epoch,
          metrics_dict_e2)

print("-"*100)
print("-"*100)
print("FINAL RESULTS:")
print("-"*100)
for key, value in metrics_dict_e2.items():
    print(f'{key}: {value}')
```

```
Train Epoch:   0 [     0/  8390 (  0%)] Loss: 2.479136  Accuracy: 6.25%
Train Epoch:   0 [  3200/  8390 ( 38%)] Loss: 2.111964  Accuracy: 16.49%
Train Epoch:   0 [  6400/  8390 ( 76%)] Loss: 2.072670  Accuracy: 18.86%
Epoch 0 - Time: 1216.79s - Train Loss: 2.162852 - Train Accuracy: 19.68%
Test Loss: 2.109093 - Test Accuracy: 22.25%

Train Epoch:   1 [     0/  8390 (  0%)] Loss: 1.894730  Accuracy: 25.00%
Train Epoch:   1 [  3200/  8390 ( 38%)] Loss: 2.064820  Accuracy: 22.62%
Train Epoch:   1 [  6400/  8390 ( 76%)] Loss: 2.095362  Accuracy: 23.46%
Epoch 1 - Time: 237.43s - Train Loss: 2.037587 - Train Accuracy: 24.28%
Test Loss: 2.066230 - Test Accuracy: 23.56%

Train Epoch:   2 [     0/  8390 (  0%)] Loss: 2.163065  Accuracy: 21.88%
Train Epoch:   2 [  3200/  8390 ( 38%)] Loss: 2.177595  Accuracy: 26.21%
Train Epoch:   2 [  6400/  8390 ( 76%)] Loss: 1.961488  Accuracy: 26.90%
Epoch 2 - Time: 3536.20s - Train Loss: 1.987517 - Train Accuracy: 26.82%
Test Loss: 2.096034 - Test Accuracy: 21.19%

Train Epoch:   3 [     0/  8390 (  0%)] Loss: 2.063694  Accuracy: 25.00%
Train Epoch:   3 [  3200/  8390 ( 38%)] Loss: 1.783233  Accuracy: 26.98%
Train Epoch:   3 [  6400/  8390 ( 76%)] Loss: 2.109796  Accuracy: 27.33%
Epoch 3 - Time: 255.05s - Train Loss: 1.977226 - Train Accuracy: 27.37%
Test Loss: 1.984386 - Test Accuracy: 26.75%

Train Epoch:   4 [     0/  8390 (  0%)] Loss: 1.770829  Accuracy: 46.88%
Train Epoch:   4 [  3200/  8390 ( 38%)] Loss: 1.907099  Accuracy: 26.30%
Train Epoch:   4 [  6400/  8390 ( 76%)] Loss: 2.030431  Accuracy: 26.40%
Epoch 4 - Time: 338.99s - Train Loss: 1.973808 - Train Accuracy: 26.90%
Test Loss: 1.946919 - Test Accuracy: 28.12%

Train Epoch:   5 [     0/  8390 (  0%)] Loss: 1.673114  Accuracy: 53.12%
Train Epoch:   5 [  3200/  8390 ( 38%)] Loss: 1.919456  Accuracy: 28.34%
Train Epoch:   5 [  6400/  8390 ( 76%)] Loss: 2.014733  Accuracy: 29.12%
Epoch 5 - Time: 249.76s - Train Loss: 1.922926 - Train Accuracy: 29.37%
Test Loss: 1.960438 - Test Accuracy: 29.12%

Train Epoch:   6 [     0/  8390 (  0%)] Loss: 1.968090  Accuracy: 25.00%
Train Epoch:   6 [  3200/  8390 ( 38%)] Loss: 1.901110  Accuracy: 29.86%
Train Epoch:   6 [  6400/  8390 ( 76%)] Loss: 2.166843  Accuracy: 30.22%
Epoch 6 - Time: 260.97s - Train Loss: 1.913745 - Train Accuracy: 30.00%
Test Loss: 2.020193 - Test Accuracy: 25.06%

Train Epoch:   7 [     0/  8390 (  0%)] Loss: 1.990562  Accuracy: 28.12%
Train Epoch:   7 [  3200/  8390 ( 38%)] Loss: 1.973588  Accuracy: 31.13%
Train Epoch:   7 [  6400/  8390 ( 76%)] Loss: 1.772776  Accuracy: 30.99%
Epoch 7 - Time: 290.29s - Train Loss: 1.895390 - Train Accuracy: 31.13%
Test Loss: 2.053939 - Test Accuracy: 31.88%

Train Epoch:   8 [     0/  8390 (  0%)] Loss: 1.993796  Accuracy: 25.00%
Train Epoch:   8 [  3200/  8390 ( 38%)] Loss: 1.942533  Accuracy: 31.50%
Train Epoch:   8 [  6400/  8390 ( 76%)] Loss: 1.756390  Accuracy: 30.89%
Epoch 8 - Time: 2350.99s - Train Loss: 1.899573 - Train Accuracy: 30.83%
Test Loss: 2.063662 - Test Accuracy: 28.69%

Train Epoch:   9 [     0/  8390 (  0%)] Loss: 1.870236  Accuracy: 43.75%
Train Epoch:   9 [  3200/  8390 ( 38%)] Loss: 1.853202  Accuracy: 31.96%
Train Epoch:   9 [  6400/  8390 ( 76%)] Loss: 1.817463  Accuracy: 32.51%
Epoch 9 - Time: 1119.26s - Train Loss: 1.858082 - Train Accuracy: 32.37%
Test Loss: 1.877559 - Test Accuracy: 30.88%
```

```
Train Epoch:  10 [     0/  8390 (  0%)] Loss: 1.855305  Accuracy: 28.12%
Train Epoch:  10 [  3200/  8390 ( 38%)] Loss: 1.746241  Accuracy: 32.09%
Train Epoch:  10 [  6400/  8390 ( 76%)] Loss: 1.742811  Accuracy: 32.70%
Epoch 10 - Time: 1714.28s - Train Loss: 1.838106 - Train Accuracy: 32.60%
Test Loss: 1.843091 - Test Accuracy: 33.31%

Train Epoch:  11 [     0/  8390 (  0%)] Loss: 1.638626  Accuracy: 46.88%
Train Epoch:  11 [  3200/  8390 ( 38%)] Loss: 1.710752  Accuracy: 34.38%
Train Epoch:  11 [  6400/  8390 ( 76%)] Loss: 1.678067  Accuracy: 33.78%
Epoch 11 - Time: 685.16s - Train Loss: 1.824078 - Train Accuracy: 33.58%
Test Loss: 1.925280 - Test Accuracy: 30.50%

Train Epoch:  12 [     0/  8390 (  0%)] Loss: 1.972635  Accuracy: 21.88%
Train Epoch:  12 [  3200/  8390 ( 38%)] Loss: 1.832804  Accuracy: 33.35%
Train Epoch:  12 [  6400/  8390 ( 76%)] Loss: 1.663738  Accuracy: 34.67%
Epoch 12 - Time: 229.54s - Train Loss: 1.800104 - Train Accuracy: 34.56%
Test Loss: 1.895443 - Test Accuracy: 29.94%

Train Epoch:  13 [     0/  8390 (  0%)] Loss: 1.694533  Accuracy: 46.88%
Train Epoch:  13 [  3200/  8390 ( 38%)] Loss: 2.048083  Accuracy: 34.31%
Train Epoch:  13 [  6400/  8390 ( 76%)] Loss: 1.721056  Accuracy: 35.49%
Epoch 13 - Time: 247.96s - Train Loss: 1.801329 - Train Accuracy: 35.18%
Test Loss: 2.059045 - Test Accuracy: 28.69%

Train Epoch:  14 [     0/  8390 (  0%)] Loss: 1.800010  Accuracy: 31.25%
Train Epoch:  14 [  3200/  8390 ( 38%)] Loss: 1.848253  Accuracy: 35.61%
Train Epoch:  14 [  6400/  8390 ( 76%)] Loss: 1.741197  Accuracy: 35.54%
Epoch 14 - Time: 228.88s - Train Loss: 1.772695 - Train Accuracy: 35.88%
Test Loss: 1.796405 - Test Accuracy: 36.19%

Train Epoch:  15 [     0/  8390 (  0%)] Loss: 1.862352  Accuracy: 34.38%
Train Epoch:  15 [  3200/  8390 ( 38%)] Loss: 1.758721  Accuracy: 36.26%
Train Epoch:  15 [  6400/  8390 ( 76%)] Loss: 1.585711  Accuracy: 36.72%
Epoch 15 - Time: 225.72s - Train Loss: 1.766506 - Train Accuracy: 36.84%
Test Loss: 1.717395 - Test Accuracy: 36.69%

Train Epoch:  16 [     0/  8390 (  0%)] Loss: 1.405506  Accuracy: 56.25%
Train Epoch:  16 [  3200/  8390 ( 38%)] Loss: 1.843312  Accuracy: 37.53%
Train Epoch:  16 [  6400/  8390 ( 76%)] Loss: 1.662048  Accuracy: 36.63%
Epoch 16 - Time: 227.74s - Train Loss: 1.762409 - Train Accuracy: 36.71%
Test Loss: 1.860578 - Test Accuracy: 30.94%

Train Epoch:  17 [     0/  8390 (  0%)] Loss: 1.506352  Accuracy: 46.88%
Train Epoch:  17 [  3200/  8390 ( 38%)] Loss: 1.742810  Accuracy: 37.41%
Train Epoch:  17 [  6400/  8390 ( 76%)] Loss: 1.585670  Accuracy: 37.00%
Epoch 17 - Time: 229.74s - Train Loss: 1.733406 - Train Accuracy: 37.16%
Test Loss: 1.890041 - Test Accuracy: 33.00%

Train Epoch:  18 [     0/  8390 (  0%)] Loss: 1.732439  Accuracy: 31.25%
Train Epoch:  18 [  3200/  8390 ( 38%)] Loss: 1.549548  Accuracy: 38.18%
Train Epoch:  18 [  6400/  8390 ( 76%)] Loss: 1.562699  Accuracy: 38.03%
Epoch 18 - Time: 236.13s - Train Loss: 1.706303 - Train Accuracy: 38.25%
Test Loss: 1.805835 - Test Accuracy: 31.38%

Train Epoch:  19 [     0/  8390 (  0%)] Loss: 1.522326  Accuracy: 53.12%
Train Epoch:  19 [  3200/  8390 ( 38%)] Loss: 1.878907  Accuracy: 39.14%
Train Epoch:  19 [  6400/  8390 ( 76%)] Loss: 1.591529  Accuracy: 38.68%
Epoch 19 - Time: 231.99s - Train Loss: 1.693270 - Train Accuracy: 39.06%
Test Loss: 1.893017 - Test Accuracy: 31.00%
```

```
Train Epoch:  20 [     0/  8390 (  0%)] Loss: 1.637941  Accuracy: 34.38%
Train Epoch:  20 [  3200/  8390 ( 38%)] Loss: 1.428763  Accuracy: 38.27%
Train Epoch:  20 [  6400/  8390 ( 76%)] Loss: 1.534950  Accuracy: 38.63%
Epoch 20 - Time: 230.55s - Train Loss: 1.689023 - Train Accuracy: 39.01%
Test Loss: 1.742904 - Test Accuracy: 35.69%

Train Epoch:  21 [     0/  8390 (  0%)] Loss: 1.540716  Accuracy: 46.88%
Train Epoch:  21 [  3200/  8390 ( 38%)] Loss: 1.768099  Accuracy: 39.29%
Train Epoch:  21 [  6400/  8390 ( 76%)] Loss: 1.482714  Accuracy: 39.23%
Epoch 21 - Time: 234.37s - Train Loss: 1.677518 - Train Accuracy: 39.01%
Test Loss: 1.751613 - Test Accuracy: 38.19%

Train Epoch:  22 [     0/  8390 (  0%)] Loss: 1.777973  Accuracy: 43.75%
Train Epoch:  22 [  3200/  8390 ( 38%)] Loss: 1.748068  Accuracy: 39.88%
Train Epoch:  22 [  6400/  8390 ( 76%)] Loss: 1.431685  Accuracy: 40.66%
Epoch 22 - Time: 234.67s - Train Loss: 1.657473 - Train Accuracy: 40.64%
Test Loss: 1.876027 - Test Accuracy: 32.38%

Train Epoch:  23 [     0/  8390 (  0%)] Loss: 2.058917  Accuracy: 25.00%
Train Epoch:  23 [  3200/  8390 ( 38%)] Loss: 1.625033  Accuracy: 38.83%
Train Epoch:  23 [  6400/  8390 ( 76%)] Loss: 1.697508  Accuracy: 39.80%
Epoch 23 - Time: 224.09s - Train Loss: 1.652329 - Train Accuracy: 40.42%
Test Loss: 1.726458 - Test Accuracy: 38.56%

Train Epoch:  24 [     0/  8390 (  0%)] Loss: 1.661963  Accuracy: 37.50%
Train Epoch:  24 [  3200/  8390 ( 38%)] Loss: 1.884388  Accuracy: 41.43%
Train Epoch:  24 [  6400/  8390 ( 76%)] Loss: 1.717796  Accuracy: 41.00%
Epoch 24 - Time: 226.88s - Train Loss: 1.633753 - Train Accuracy: 41.20%
Test Loss: 1.755266 - Test Accuracy: 39.50%

Train Epoch:  25 [     0/  8390 (  0%)] Loss: 2.029453  Accuracy: 21.88%
Train Epoch:  25 [  3200/  8390 ( 38%)] Loss: 1.803970  Accuracy: 41.62%
Train Epoch:  25 [  6400/  8390 ( 76%)] Loss: 1.742526  Accuracy: 41.48%
Epoch 25 - Time: 241.10s - Train Loss: 1.638339 - Train Accuracy: 41.25%
Test Loss: 1.831373 - Test Accuracy: 35.31%

Train Epoch:  26 [     0/  8390 (  0%)] Loss: 1.667392  Accuracy: 43.75%
Train Epoch:  26 [  3200/  8390 ( 38%)] Loss: 1.706036  Accuracy: 43.75%
Train Epoch:  26 [  6400/  8390 ( 76%)] Loss: 1.283091  Accuracy: 41.93%
Epoch 26 - Time: 242.64s - Train Loss: 1.636386 - Train Accuracy: 41.80%
Test Loss: 1.872317 - Test Accuracy: 37.25%

Train Epoch:  27 [     0/  8390 (  0%)] Loss: 1.798654  Accuracy: 37.50%
Train Epoch:  27 [  3200/  8390 ( 38%)] Loss: 1.548966  Accuracy: 40.69%
Train Epoch:  27 [  6400/  8390 ( 76%)] Loss: 1.751830  Accuracy: 42.37%
Epoch 27 - Time: 227.35s - Train Loss: 1.597866 - Train Accuracy: 42.66%
Test Loss: 1.769318 - Test Accuracy: 35.69%

Train Epoch:  28 [     0/  8390 (  0%)] Loss: 1.510158  Accuracy: 40.62%
Train Epoch:  28 [  3200/  8390 ( 38%)] Loss: 1.709183  Accuracy: 43.53%
Train Epoch:  28 [  6400/  8390 ( 76%)] Loss: 1.747401  Accuracy: 42.75%
Epoch 28 - Time: 226.97s - Train Loss: 1.610359 - Train Accuracy: 42.72%
Test Loss: 1.805375 - Test Accuracy: 37.50%

Train Epoch:  29 [     0/  8390 (  0%)] Loss: 1.620673  Accuracy: 43.75%
Train Epoch:  29 [  3200/  8390 ( 38%)] Loss: 1.735744  Accuracy: 43.16%
Train Epoch:  29 [  6400/  8390 ( 76%)] Loss: 1.518578  Accuracy: 43.52%
Epoch 29 - Time: 222.71s - Train Loss: 1.580171 - Train Accuracy: 43.17%
Test Loss: 1.713054 - Test Accuracy: 37.12%
```

```
Train Epoch:  30 [     0/  8390 (  0%)] Loss: 1.523651  Accuracy: 43.75%
Train Epoch:  30 [  3200/  8390 ( 38%)] Loss: 1.402524  Accuracy: 42.82%
Train Epoch:  30 [  6400/  8390 ( 76%)] Loss: 1.423656  Accuracy: 43.17%
Epoch 30 - Time: 221.71s - Train Loss: 1.588955 - Train Accuracy: 43.31%
Test Loss: 1.720405 - Test Accuracy: 40.56%

Train Epoch:  31 [     0/  8390 (  0%)] Loss: 1.578195  Accuracy: 53.12%
Train Epoch:  31 [  3200/  8390 ( 38%)] Loss: 1.480698  Accuracy: 45.45%
Train Epoch:  31 [  6400/  8390 ( 76%)] Loss: 1.441470  Accuracy: 45.94%
Epoch 31 - Time: 224.97s - Train Loss: 1.546672 - Train Accuracy: 45.16%
Test Loss: 1.775559 - Test Accuracy: 38.19%

Train Epoch:  32 [     0/  8390 (  0%)] Loss: 1.695147  Accuracy: 37.50%
Train Epoch:  32 [  3200/  8390 ( 38%)] Loss: 1.325187  Accuracy: 44.86%
Train Epoch:  32 [  6400/  8390 ( 76%)] Loss: 1.499307  Accuracy: 44.71%
Epoch 32 - Time: 232.97s - Train Loss: 1.528269 - Train Accuracy: 45.01%
Test Loss: 1.851925 - Test Accuracy: 35.56%

Train Epoch:  33 [     0/  8390 (  0%)] Loss: 1.566870  Accuracy: 37.50%
Train Epoch:  33 [  3200/  8390 ( 38%)] Loss: 1.623017  Accuracy: 46.81%
Train Epoch:  33 [  6400/  8390 ( 76%)] Loss: 1.333343  Accuracy: 46.13%
Epoch 33 - Time: 222.50s - Train Loss: 1.519943 - Train Accuracy: 45.77%
Test Loss: 1.797330 - Test Accuracy: 38.19%

Train Epoch:  34 [     0/  8390 (  0%)] Loss: 1.530742  Accuracy: 53.12%
Train Epoch:  34 [  3200/  8390 ( 38%)] Loss: 1.340442  Accuracy: 46.26%
Train Epoch:  34 [  6400/  8390 ( 76%)] Loss: 1.714650  Accuracy: 46.25%
Epoch 34 - Time: 371.90s - Train Loss: 1.506658 - Train Accuracy: 46.63%
Test Loss: 1.941669 - Test Accuracy: 35.25%

Train Epoch:  35 [     0/  8390 (  0%)] Loss: 1.191646  Accuracy: 65.62%
Train Epoch:  35 [  3200/  8390 ( 38%)] Loss: 1.346883  Accuracy: 46.75%
Train Epoch:  35 [  6400/  8390 ( 76%)] Loss: 1.417341  Accuracy: 46.50%
Epoch 35 - Time: 839.33s - Train Loss: 1.492409 - Train Accuracy: 46.16%
Test Loss: 1.755629 - Test Accuracy: 39.75%

Train Epoch:  36 [     0/  8390 (  0%)] Loss: 1.476019  Accuracy: 46.88%
Train Epoch:  36 [  3200/  8390 ( 38%)] Loss: 1.865099  Accuracy: 46.26%
Train Epoch:  36 [  6400/  8390 ( 76%)] Loss: 1.578250  Accuracy: 47.51%
Epoch 36 - Time: 260.74s - Train Loss: 1.480662 - Train Accuracy: 47.54%
Test Loss: 1.746769 - Test Accuracy: 41.62%

Train Epoch:  37 [     0/  8390 (  0%)] Loss: 1.372396  Accuracy: 65.62%
Train Epoch:  37 [  3200/  8390 ( 38%)] Loss: 1.404371  Accuracy: 48.58%
Train Epoch:  37 [  6400/  8390 ( 76%)] Loss: 1.312795  Accuracy: 47.89%
Epoch 37 - Time: 238.93s - Train Loss: 1.456575 - Train Accuracy: 47.79%
Test Loss: 1.713381 - Test Accuracy: 40.44%

Train Epoch:  38 [     0/  8390 (  0%)] Loss: 1.285304  Accuracy: 46.88%
Train Epoch:  38 [  3200/  8390 ( 38%)] Loss: 1.466446  Accuracy: 47.06%
Train Epoch:  38 [  6400/  8390 ( 76%)] Loss: 1.480421  Accuracy: 47.50%
Epoch 38 - Time: 225.90s - Train Loss: 1.466263 - Train Accuracy: 47.52%
Test Loss: 1.734649 - Test Accuracy: 39.38%

Train Epoch:  39 [     0/  8390 (  0%)] Loss: 1.281939  Accuracy: 56.25%
Train Epoch:  39 [  3200/  8390 ( 38%)] Loss: 1.203103  Accuracy: 49.04%
Train Epoch:  39 [  6400/  8390 ( 76%)] Loss: 1.221993  Accuracy: 48.69%
Epoch 39 - Time: 233.83s - Train Loss: 1.438474 - Train Accuracy: 48.75%
Test Loss: 1.698388 - Test Accuracy: 41.25%
```

```
Train Epoch:  40 [     0/  8390 (  0%)] Loss: 1.390738  Accuracy: 50.00%
Train Epoch:  40 [  3200/  8390 ( 38%)] Loss: 1.468864  Accuracy: 48.05%
Train Epoch:  40 [  6400/  8390 ( 76%)] Loss: 1.655652  Accuracy: 49.02%
Epoch 40 - Time: 234.83s - Train Loss: 1.442420 - Train Accuracy: 48.86%
Test Loss: 1.713381 - Test Accuracy: 40.88%

Train Epoch:  41 [     0/  8390 (  0%)] Loss: 1.223912  Accuracy: 46.88%
Train Epoch:  41 [  3200/  8390 ( 38%)] Loss: 1.173113  Accuracy: 48.42%
Train Epoch:  41 [  6400/  8390 ( 76%)] Loss: 1.541021  Accuracy: 48.74%
Epoch 41 - Time: 240.24s - Train Loss: 1.412495 - Train Accuracy: 48.99%
Test Loss: 1.804260 - Test Accuracy: 40.00%

Train Epoch:  42 [     0/  8390 (  0%)] Loss: 1.315914  Accuracy: 56.25%
Train Epoch:  42 [  3200/  8390 ( 38%)] Loss: 1.457621  Accuracy: 50.93%
Train Epoch:  42 [  6400/  8390 ( 76%)] Loss: 1.301119  Accuracy: 49.56%
Epoch 42 - Time: 228.13s - Train Loss: 1.408532 - Train Accuracy: 49.81%
Test Loss: 1.848636 - Test Accuracy: 36.69%

Train Epoch:  43 [     0/  8390 (  0%)] Loss: 1.095912  Accuracy: 68.75%
Train Epoch:  43 [  3200/  8390 ( 38%)] Loss: 1.502681  Accuracy: 52.04%
Train Epoch:  43 [  6400/  8390 ( 76%)] Loss: 1.337117  Accuracy: 50.92%
Epoch 43 - Time: 231.53s - Train Loss: 1.396336 - Train Accuracy: 50.35%
Test Loss: 1.700084 - Test Accuracy: 42.94%

Train Epoch:  44 [     0/  8390 (  0%)] Loss: 1.331070  Accuracy: 53.12%
Train Epoch:  44 [  3200/  8390 ( 38%)] Loss: 1.648317  Accuracy: 51.79%
Train Epoch:  44 [  6400/  8390 ( 76%)] Loss: 1.354382  Accuracy: 51.85%
Epoch 44 - Time: 234.37s - Train Loss: 1.361948 - Train Accuracy: 51.42%
Test Loss: 1.661956 - Test Accuracy: 42.50%

Train Epoch:  45 [     0/  8390 (  0%)] Loss: 1.314737  Accuracy: 62.50%
Train Epoch:  45 [  3200/  8390 ( 38%)] Loss: 1.620541  Accuracy: 53.00%
Train Epoch:  45 [  6400/  8390 ( 76%)] Loss: 1.276034  Accuracy: 51.38%
Epoch 45 - Time: 237.77s - Train Loss: 1.358715 - Train Accuracy: 51.16%
Test Loss: 1.723875 - Test Accuracy: 39.81%

Train Epoch:  46 [     0/  8390 (  0%)] Loss: 1.273015  Accuracy: 53.12%
Train Epoch:  46 [  3200/  8390 ( 38%)] Loss: 1.205081  Accuracy: 51.89%
Train Epoch:  46 [  6400/  8390 ( 76%)] Loss: 1.084803  Accuracy: 51.94%
Epoch 46 - Time: 230.29s - Train Loss: 1.345437 - Train Accuracy: 52.11%
Test Loss: 1.786524 - Test Accuracy: 40.50%

Train Epoch:  47 [     0/  8390 (  0%)] Loss: 1.356501  Accuracy: 46.88%
Train Epoch:  47 [  3200/  8390 ( 38%)] Loss: 1.344147  Accuracy: 52.26%
Train Epoch:  47 [  6400/  8390 ( 76%)] Loss: 1.232960  Accuracy: 52.39%
Epoch 47 - Time: 235.62s - Train Loss: 1.351638 - Train Accuracy: 52.31%
Test Loss: 1.924757 - Test Accuracy: 37.25%

Train Epoch:  48 [     0/  8390 (  0%)] Loss: 1.029061  Accuracy: 65.62%
Train Epoch:  48 [  3200/  8390 ( 38%)] Loss: 1.407160  Accuracy: 52.13%
Train Epoch:  48 [  6400/  8390 ( 76%)] Loss: 1.230066  Accuracy: 52.91%
Epoch 48 - Time: 237.75s - Train Loss: 1.333594 - Train Accuracy: 53.11%
Test Loss: 1.675849 - Test Accuracy: 42.44%

Train Epoch:  49 [     0/  8390 (  0%)] Loss: 1.518033  Accuracy: 46.88%
Train Epoch:  49 [  3200/  8390 ( 38%)] Loss: 1.227060  Accuracy: 52.78%
Train Epoch:  49 [  6400/  8390 ( 76%)] Loss: 1.245394  Accuracy: 53.25%
Epoch 49 - Time: 228.26s - Train Loss: 1.321129 - Train Accuracy: 53.13%
Test Loss: 1.728029 - Test Accuracy: 41.00%
```

```
--------------------------------------------------------------------------------------------------------
--------------------------------------------------------------------------------------------------------
FINAL RESULTS:
--------------------------------------------------------------------------------------------------------
epoch_times: [1216.786159992218, 237.43125772476196, 3536.2034180164337, 2
55.0527946949005, 338.98512411117554, 249.76298189163208, 260.968131065368
65, 290.2906048297882, 2350.9871430397034, 1119.262617111206, 1714.2808780
670166, 685.1609690189362, 229.54470205307007, 247.95893597602844, 228.884
4177722931, 225.72268295288086, 227.7428710460628, 229.74079060554504, 23
6.1341609954834, 231.98818516731262, 230.55307173728943, 234.3666918277740
5, 234.67452025413513, 224.09229612350464, 226.87726593017578, 241.1005702
0187378, 242.64449501037598, 227.35298085212708, 226.97432398796082, 222.7
1014785766602, 221.70982909202576, 224.96943998336792, 232.96854400634766,
222.49809098243713, 371.89834690093994, 839.3271780014038, 260.74057102203
37, 238.9295699596405, 225.8984100818634, 233.8280041217804, 234.825166702
2705, 240.2430510520935, 228.12757992744446, 231.5294008255005, 234.373477
22053528, 237.76521015167236, 230.28806805610657, 235.62296104431152, 237.
7456498146057, 228.25890517234802]
train_losses: [2.162851532906976, 2.0375869633587262, 1.9875167171463712,
1.9772260184506423, 1.973807847226849, 1.9229257152280734, 1.9137453100153
508, 1.8953898330681196, 1.899573454420075, 1.8580818735916196, 1.83810602
11713078, 1.824077689920673, 1.8001040738957528, 1.8013294184480915, 1.772
6950404298214, 1.7665060276293572, 1.7624088416572745, 1.7334060755394798,
1.706302819816211, 1.6932695644502422, 1.6890230351731976, 1.6775176966463
337, 1.657472894846938, 1.6523291555069786, 1.6337526204021833, 1.63833934
61467655, 1.6363864782202335, 1.5978663550078414, 1.6103589298160932, 1.58
0170889392154, 1.588955131650881, 1.546672313267948, 1.5282688573116565,
1.5199425070340398, 1.506657939375812, 1.492409292978185, 1.48066221894198
7, 1.4565749753067512, 1.4662629063347823, 1.4384736666697582, 1.442420180
5172985, 1.4124947722631556, 1.408531955407776, 1.3963362629177005, 1.3619
47868844025, 1.358714550961065, 1.3454369156415227, 1.3516383494129618, 1.
3335936713309688, 1.3211287958476379]
train_accuracies: [19.67818831942789, 24.27890345649583, 26.81764004767580
5, 27.36591179976162, 26.90107270560191, 29.36829558998808, 30.0, 31.13230
0357568536, 30.834326579261024, 32.37187127532777, 32.598331346841476, 33.
57568533969011, 34.56495828367104, 35.18474374255066, 35.87604290822408, 3
6.841477949940405, 36.7103694874851, 37.163289630512516, 38.2479141835518
5, 39.05840286054827, 39.01072705601907, 39.01072705601907, 40.64362336114
422, 40.417163289630516, 41.203814064362334, 41.25148986889154, 41.7997616
2097736, 42.65792610250298, 42.717520858164484, 43.1704410011919, 43.31346
84147795, 45.160905840286055, 45.00595947556615, 45.76877234803337, 46.626
936829559, 46.162097735399286, 47.54469606674613, 47.794994040524436, 47.5
20858164481524, 48.74851013110846, 48.85578069129917, 48.98688915375447, 4
9.80929678188319, 50.345649582836714, 51.418355184743746, 51.1561382598331
35, 52.109654350417166, 52.31227651966627, 53.1108462455304, 53.1346841477
9499]
test_losses: [2.109093050956726, 2.066229785680771, 2.0960341399908065, 1.
9843857073783875, 1.9469191420078278, 1.9604377828538417, 2.02019314527511
6, 2.0539393174648284, 2.0636624544113875, 1.877559413909912, 1.8430911654
233932, 1.92527961820364, 1.8954430627822876, 2.059045124053955, 1.7964048
707485198, 1.717394924610853, 1.8605775427818299, 1.890041069984436, 1.80
5834757089615, 1.893017338514328, 1.7429044938087463, 1.751613089442253,
1.8760265758633614, 1.7264579290151596, 1.7552662622928619, 1.831373276114
4639, 1.87231743901968, 1.7693175518512725, 1.805375024974346, 1.713054386
973381, 1.7204049178957939, 1.7755592098832131, 1.8519245845079422, 1.7973
303461819887, 1.9416690100729466, 1.755628821849823, 1.7467694664001465,
1.7133814676105976, 1.734649378284812, 1.698387702703476, 1.71338147729635
```

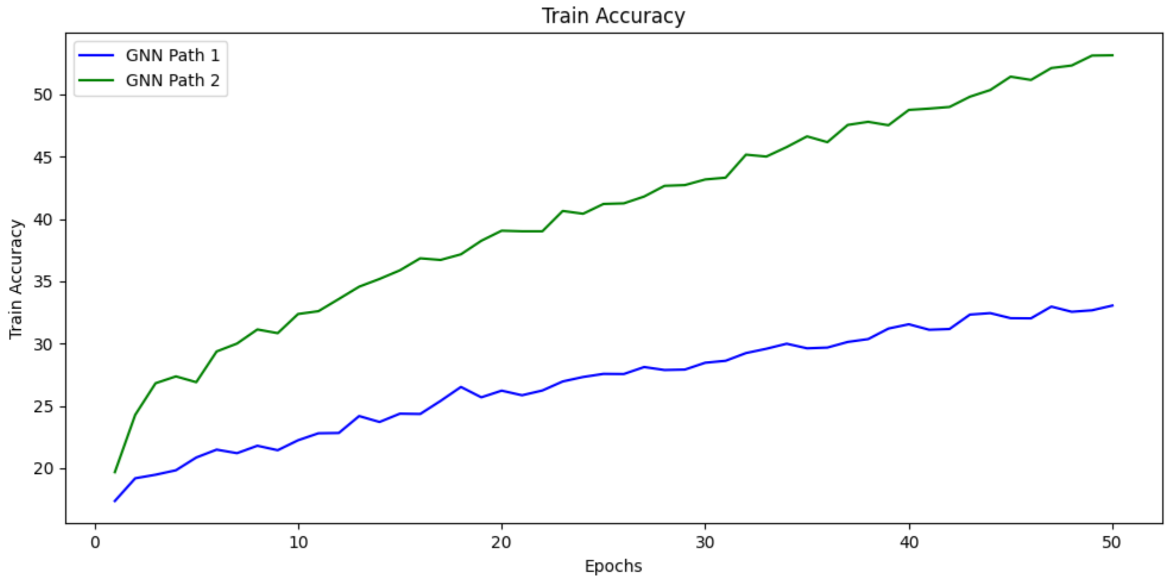
```
25, 1.8042599228024483, 1.8486363255977631, 1.7000837564468383, 1.66195631
92129134, 1.7238749971985816, 1.7865237033367156, 1.9247570157051086, 1.67
58494615554809, 1.7280293914675713]
test_accuracies: [22.25, 23.5625, 21.1875, 26.75, 28.125, 29.125, 25.0625,
31.875, 28.6875, 30.875, 33.3125, 30.5, 29.9375, 28.6875, 36.1875, 36.687
5, 30.9375, 33.0, 31.375, 31.0, 35.6875, 38.1875, 32.375, 38.5625, 39.5, 3
5.3125, 37.25, 35.6875, 37.5, 37.125, 40.5625, 38.1875, 35.5625, 38.1875,
35.25, 39.75, 41.625, 40.4375, 39.375, 41.25, 40.875, 40.0, 36.6875, 42.93
75, 42.5, 39.8125, 40.5, 37.25, 42.4375, 41.0]
```

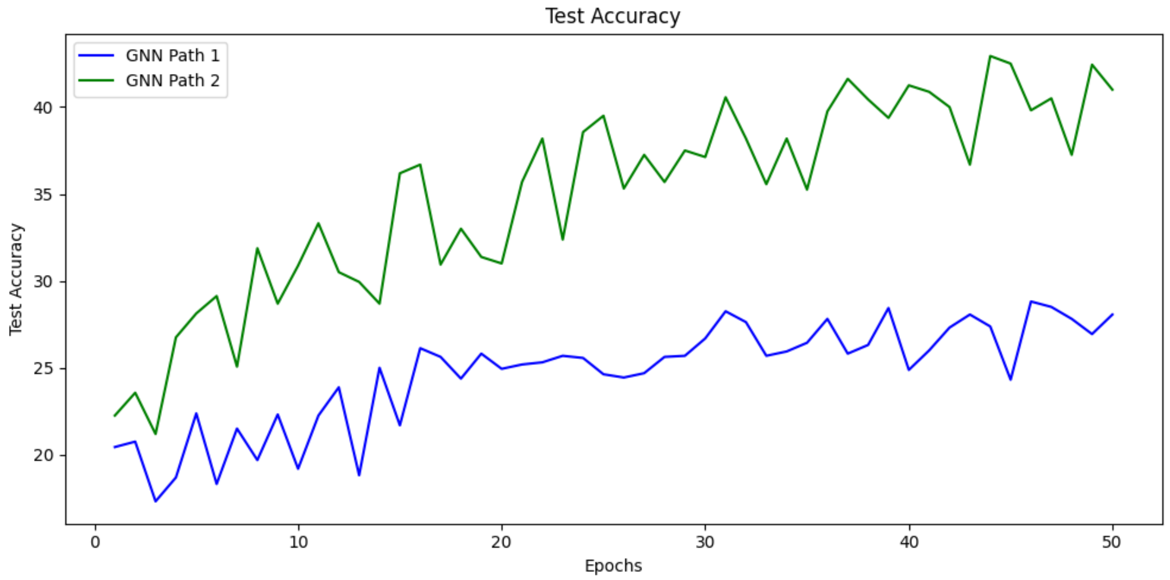
Plots

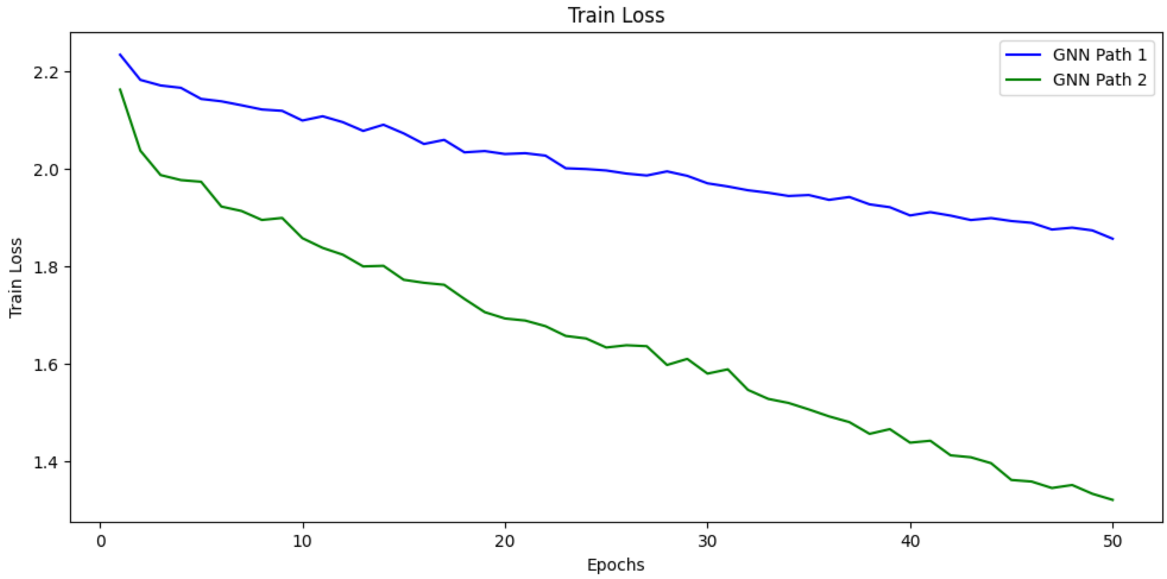
In [40]:

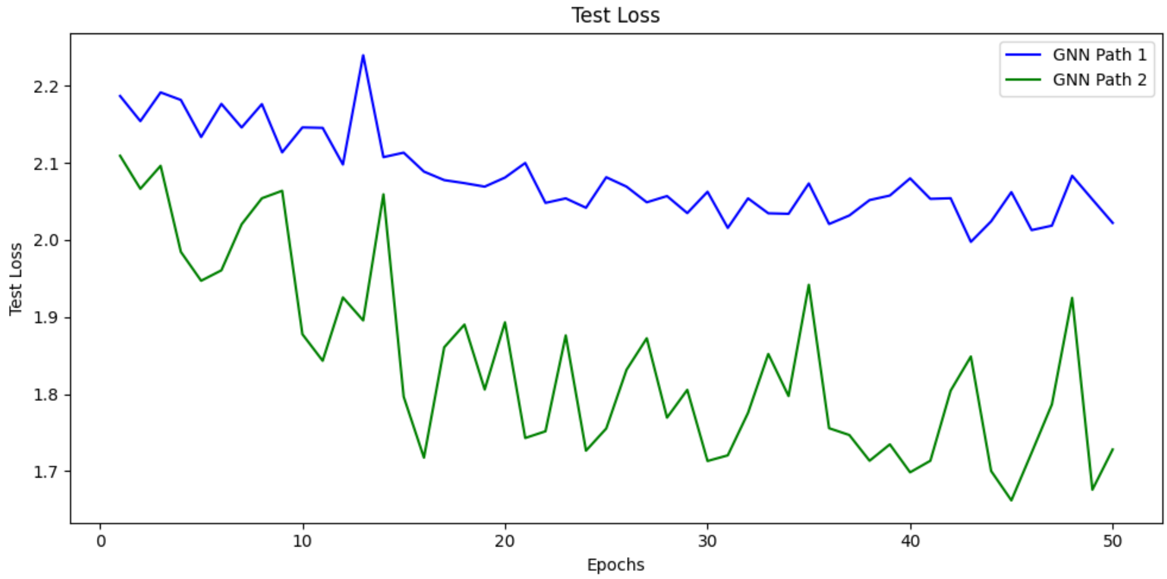
```python
# Data for the four scenarios
data = {
    "GNN Path 1": metrics_dict_e1,
    "GNN Path 2": metrics_dict_e2,
}

# Data for plotting
epochs = range(1, 51)
colors = ['b', 'g', 'r', 'm', 'y', 'c', 'k', '#FF5733', '#7E4DFF']
scenarios = list(data.keys())

fig, axes = plt.subplots(2, 1, figsize=(10, 10))

for i, scenario in enumerate(scenarios):
    axes[0].plot(epochs, data[scenario]["train_accuracies"], label=scenar

axes[0].set_title("Train Accuracy")
axes[0].set_xlabel("Epochs")
axes[0].set_ylabel("Train Accuracy")
axes[0].legend()

for i, scenario in enumerate(scenarios):
    axes[1].plot(epochs, data[scenario]["test_accuracies"], label=scenari

axes[1].set_title("Test Accuracy")
axes[1].set_xlabel("Epochs")
axes[1].set_ylabel("Test Accuracy")
axes[1].legend()

plt.tight_layout()
plt.show()

fig, axes = plt.subplots(2, 1, figsize=(10, 10))

for i, scenario in enumerate(scenarios):
    axes[0].plot(epochs, data[scenario]["train_losses"], label=scenario, 

axes[0].set_title("Train Loss")
axes[0].set_xlabel("Epochs")
axes[0].set_ylabel("Train Loss")
axes[0].legend()

for i, scenario in enumerate(scenarios):
    axes[1].plot(epochs, data[scenario]["test_losses"], label=scenario, c

axes[1].set_title("Test Loss")
```

```
axes[1].set_xlabel("Epochs")
axes[1].set_ylabel("Test Loss")
axes[1].legend()

plt.tight_layout()
plt.show()

fig, axes = plt.subplots(1, 1, figsize=(10, 5))
for i, scenario in enumerate(scenarios):
    axes.plot(epochs, data[scenario]["epoch_times"], label=scenario, colc
axes.set_title("Time")
axes.set_xlabel("Epochs")
axes.set_ylabel("Time (secs)")
axes.legend()
```









Out[40]: <matplotlib.legend.Legend at 0x2af840c50>