code

December 22, 2021

[ ]:
```
import numpy as np
import torch
import os
from sklearn import svm
import pywt
import matplotlib.pyplot as plt
plt.rcParams.update({'font.size': 22})
from sklearn.manifold import TSNE
import torch.nn as nn
from torch.nn.functional import softmax, relu
from torch.utils.data import DataLoader, TensorDataset
from tqdm import tqdm
from sklearn.metrics import accuracy_score

path_figures = './figures'
if not os.path.exists(path_figures):
    os.makedirs(path_figures)

path_model = './models'
if not os.path.exists(path_model):
    os.makedirs(path_model)
```

load dataset and check data shape

[ ]:
```
data_path = './data'
X_train_valid = np.load(os.path.join(data_path, 'X_train_valid.npy'))
y_train_valid = np.load(os.path.join(data_path, 'y_train_valid.npy'))
X_test = np.load(os.path.join(data_path, 'X_test.npy'))
y_test = np.load(os.path.join(data_path, 'y_test.npy'))
person_train_valid = np.load(os.path.join(data_path, 'person_train_valid.npy'))
person_test = np.load(os.path.join(data_path, 'person_test.npy'))

print(X_train_valid.shape)
print(y_train_valid.shape)
print(X_test.shape)
print(y_test.shape)
print(person_train_valid.shape)
print(person_test.shape)
```

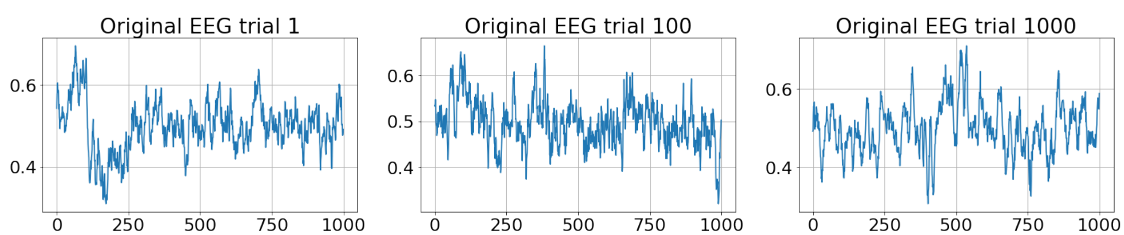
```
def func_categorical_label(y):
    y[y == 769] = 0
    y[y == 770] = 1
    y[y == 771] = 2
    y[y == 772] = 3
    return y

# minmax normalization
def func_normalize(x):
    return (x - np.min(x))/(np.max(x) - np.min(x))

X_train_valid = func_normalize(X_train_valid)
X_test = func_normalize(X_test)

# change the label from subject index (769, ...) to (0, 1, 2, 3)
y_train_valid = func_categorical_label(y_train_valid)
y_test = func_categorical_label(y_test)
```

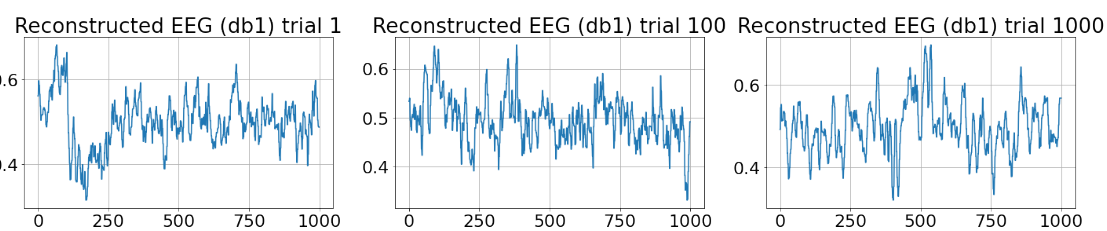
```
(2115, 22, 1000)
(2115,)
(443, 22, 1000)
(443,)
(2115, 1)
(443, 1)
```

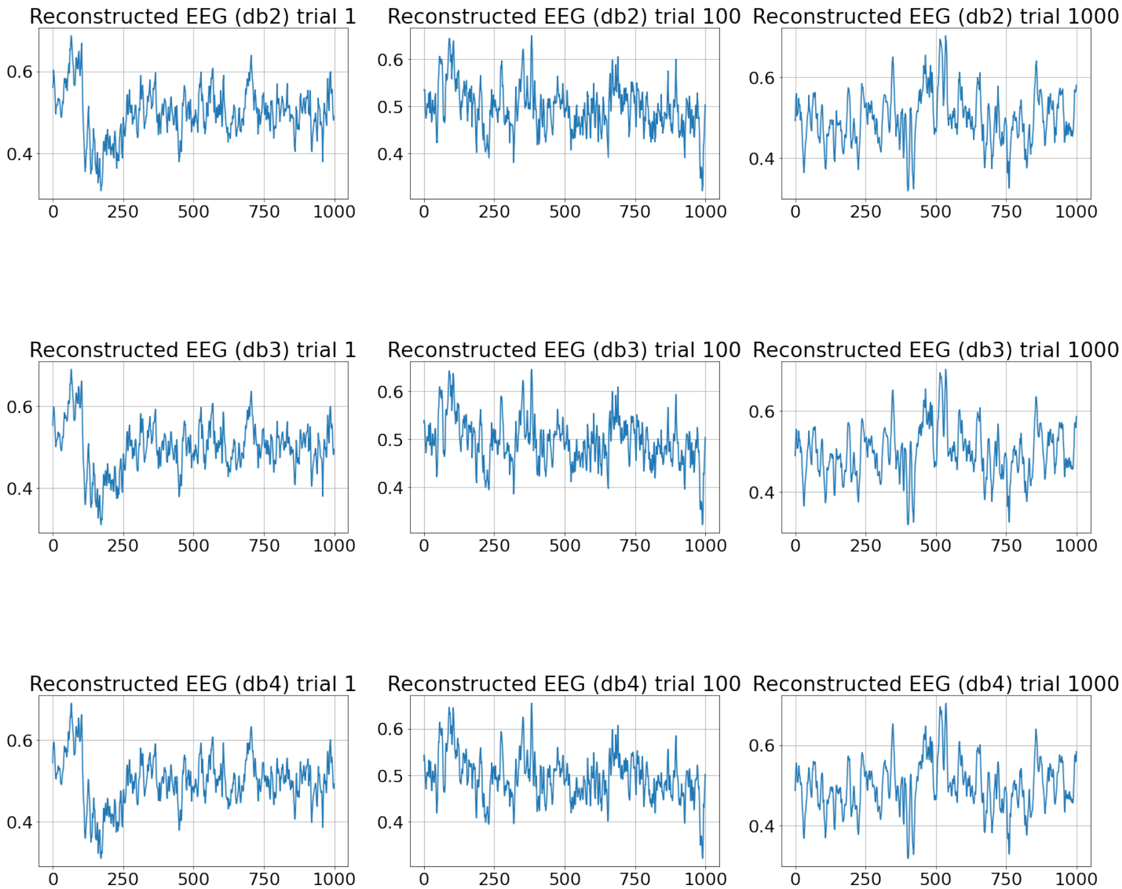
Compute wavelet transform embedding with various levels

[ ]:
```
def func_plotEEG(X, trial_list, fig_name, path_figures, subplot_title):
    # plot EEG signal according to the trial list
    numOfSample = len(trial_list)
    time_step = X.shape[2]
    fig, axes = plt.subplots(1, numOfSample)
    fig.set_size_inches(8*numOfSample,4)
    for i in range(numOfSample):
        axes[i].plot(np.arange(time_step), X[trial_list[i], 0, :]) # only plot␣
 ↪the first node
        axes[i].grid(True)
        axes[i].title.set_text(subplot_title + ' trial '+ str(trial_list[i]))
    fig.savefig(os.path.join(path_figures, fig_name), bbox_inches='tight')

def func_fitsvm(X, y):
    X = np.reshape(X, (X.shape[0], X.shape[1]*X.shape[2]))
    clf = svm.SVC()
    clf.fit(X, y)
    return clf

def func_dwt(X, db_level):
```

```
    (cA, cD) = pywt.dwt(X, db_level)
    recon_X_cA = pywt.idwt(cA, None, db_level)
    recon_X_cD = pywt.idwt(cD, None, db_level)
    return cA, cD, recon_X_cA, recon_X_cD

cA_db1, _, recon_db1_X_train_valid, _ = func_dwt(X_train_valid, 'db1')
cA_db2, _, recon_db2_X_train_valid, _ = func_dwt(X_train_valid, 'db2')
cA_db3, _, recon_db3_X_train_valid, _ = func_dwt(X_train_valid, 'db3')
cA_db4, _, recon_db4_X_train_valid, _ = func_dwt(X_train_valid, 'db4')

trial_list = [1,100,1000]
func_plotEEG(X_train_valid, trial_list, 'org_X_train_valid.pdf', path_figures,␣
 ↪'Original EEG')
func_plotEEG(recon_db1_X_train_valid, trial_list, 'recon_db1_X_train_valid.
 ↪pdf', path_figures, 'Reconstructed EEG (db1)')
func_plotEEG(recon_db2_X_train_valid, trial_list, 'recon_db2_X_train_valid.
 ↪pdf', path_figures, 'Reconstructed EEG (db2)')
func_plotEEG(recon_db3_X_train_valid, trial_list, 'recon_db3_X_train_valid.
 ↪pdf', path_figures, 'Reconstructed EEG (db3)')
func_plotEEG(recon_db4_X_train_valid, trial_list, 'recon_db4_X_train_valid.
 ↪pdf', path_figures, 'Reconstructed EEG (db4)')

svm_db1 = func_fitsvm(cA_db1, y_train_valid)
svm_db2 = func_fitsvm(cA_db2, y_train_valid)
svm_db3 = func_fitsvm(cA_db3, y_train_valid)
svm_db4 = func_fitsvm(cA_db4, y_train_valid)
```







[ ]:
```
test_cA_db1, _, recon_db1_X_test, _ = func_dwt(X_test, 'db1')
test_cA_db2, _, recon_db2_X_test, _ = func_dwt(X_test, 'db2')
test_cA_db3, _, recon_db3_X_test, _ = func_dwt(X_test, 'db3')
test_cA_db4, _, recon_db4_X_test, _ = func_dwt(X_test, 'db4')

def func_testReshape(X):
    return np.reshape(X, (X.shape[0],X.shape[1]*X.shape[2]))

pred_db1 = svm_db1.predict(func_testReshape(test_cA_db1))
pred_db2 = svm_db2.predict(func_testReshape(test_cA_db2))
pred_db3 = svm_db3.predict(func_testReshape(test_cA_db3))
pred_db4 = svm_db4.predict(func_testReshape(test_cA_db4))
```

[ ]:
```
print('wavelet db1 accuracy:', accuracy_score(pred_db1, y_test))
print('wavelet db2 accuracy:', accuracy_score(pred_db2, y_test))
print('wavelet db3 accuracy:', accuracy_score(pred_db3, y_test))
print('wavelet db4 accuracy:', accuracy_score(pred_db4, y_test))
```

```
wavelet db1 accuracy: 0.3927765237020316
wavelet db2 accuracy: 0.3927765237020316
```

```
wavelet db3 accuracy: 0.39503386004514673
wavelet db4 accuracy: 0.3837471783295711
```

Compute TSNE embedding

[ ]:
```python
# compute the tsne-embedding and use SVM for feature selection
tsne_train_valid = TSNE(n_components=3).fit_transform(np.reshape(X_train_valid,␣
↪(X_train_valid.shape[0], X_train_valid.shape[1]*X_train_valid.shape[2])))
tsne_test = TSNE(n_components=3).fit_transform(np.reshape(X_test, (X_test.
↪shape[0], X_test.shape[1]*X_test.shape[2])))

svm_tsne = svm.SVC()
svm_tsne.fit(tsne_train_valid, y_train_valid)
pred_tsne = svm_tsne.predict(tsne_test)

print('tsne accuracy', accuracy_score(pred_tsne, y_test))
```

```
/home/xiaoranzhang/anaconda3/envs/torch_env/lib/python3.9/site-
packages/sklearn/manifold/_t_sne.py:780: FutureWarning: The default
initialization in TSNE will change from 'random' to 'pca' in 1.2.
  warnings.warn(
/home/xiaoranzhang/anaconda3/envs/torch_env/lib/python3.9/site-
packages/sklearn/manifold/_t_sne.py:790: FutureWarning: The default learning
rate in TSNE will change from 200.0 to 'auto' in 1.2.
  warnings.warn(
/home/xiaoranzhang/anaconda3/envs/torch_env/lib/python3.9/site-
packages/sklearn/manifold/_t_sne.py:780: FutureWarning: The default
initialization in TSNE will change from 'random' to 'pca' in 1.2.
  warnings.warn(
/home/xiaoranzhang/anaconda3/envs/torch_env/lib/python3.9/site-
packages/sklearn/manifold/_t_sne.py:790: FutureWarning: The default learning
rate in TSNE will change from 200.0 to 'auto' in 1.2.
  warnings.warn(
```

```
tsne accuracy 0.2618510158013544
```

[ ]:
```python
# implement the neural network embedding
def evaluate(model, validation_set, loss_fn):
    with torch.no_grad():
        numOfCorrectLabels = 0
        total_loss = 0
        for data, label in validation_set:
            model_input = torch.reshape(data, (data.shape[0], 1, data.shape[1],␣
↪data.shape[2])) # expand one dimension for the channel for the EGG
            model_input = model_input.to('cuda')
            label = label.to('cuda')

            pred_onehot, x_embd = model(model_input)
```

```python
            pred = torch.argmax(pred_onehot, dim=1)

            eval_loss = loss_fn(pred_onehot, label.to(torch.long))

            total_loss += eval_loss.item()
            numOfCorrectLabels += (pred == label).float().sum()

        accuracy = numOfCorrectLabels / len(validation_set.dataset)
        total_loss /= len(validation_set.dataset)
    return accuracy, total_loss

class ShallowCNN(nn.Module):
    """ Simple feed forward network with one hidden layer."""
    def __init__(self):
        super(ShallowCNN, self).__init__()
        self.conv1 = nn.Conv2d(1, 40, (1, 25), padding='valid')
        self.conv2 = nn.Conv2d(1, 40, (880, 1), padding='valid')
        self.pool1 = nn.AvgPool2d(kernel_size=(1,75), stride=(1, 15))
        self.ln1 = nn.Linear(2440, 4)

    def forward(self, x):
        x = self.conv1(x)
        # print(x.shape) # bs x 40 x 22 x 976
        x = torch.reshape(x, (x.shape[0], 1, 880, 976))
        x = relu(self.conv2(x))
        # print(x.shape) # bs x 40 x 1 x 976
        x = torch.reshape(x, (x.shape[0], 40, 976))
        x = relu(self.pool1(x))
        x_embd = torch.flatten(x, start_dim=1) # flatten the batched array
        # print(x_embd.shape) # bs x 2440
        pred = softmax(self.ln1(x_embd))
        # print(x.shape) # bs x 4

        return pred, x_embd

# create train and validation set for shallow CNN
n_split = 0.8
bs = 4
num_epoch = 1000
lr = 1e-4
numOfSample = X_train_valid.shape[0]


X_train = torch.Tensor(X_train_valid[:int(n_split*numOfSample)])
y_train = torch.Tensor(y_train_valid[:int(n_split*numOfSample)])
X_valid = torch.Tensor(X_train_valid[int(n_split*numOfSample):])
y_valid = torch.Tensor(y_train_valid[int(n_split*numOfSample):])
```

```python
EEG_train_dataset = DataLoader(TensorDataset(X_train, y_train), batch_size=bs,␣
↪shuffle=True)
EEG_valid_dataset = DataLoader(TensorDataset(X_valid, y_valid), batch_size=bs,␣
↪shuffle=True)

print('Training input shape', X_train.shape)
print('Validation input shape', X_valid.shape)

model = ShallowCNN()
model.to('cuda')
CE_loss = nn.CrossEntropyLoss()
optimizer = torch.optim.Adam(model.parameters(), lr=lr)

# start training
base_valid_acc = 0
train_acc_list, train_loss_list, valid_acc_list, valid_loss_list = [], [], [],␣
↪[]
for epoch_idx in range(num_epoch):
    for data, label in EEG_train_dataset:
        optimizer.zero_grad()
        model_input = torch.reshape(data, (data.shape[0], 1, data.shape[1],␣
↪data.shape[2])) # expand one dimension for the channel for the EGG
        model_input = model_input.to('cuda')
        label = label.to('cuda')

        pred_onehot, x_embd = model(model_input)
        pred = torch.argmax(pred_onehot, dim=1)

        loss = CE_loss(pred_onehot.to(torch.float32), label.to(torch.long)) #␣
↪pred onhot, label just digit (long tensor!!!)

        # print(one_hot(label.to(torch.int64)).to(torch.float32).shape)
        loss.backward()

        optimizer.step()

    if epoch_idx % 10 == 0:
        train_acc, train_loss = evaluate(model, EEG_train_dataset, CE_loss)
        valid_acc, valid_loss = evaluate(model, EEG_valid_dataset, CE_loss)
        train_acc_list.append(train_acc), valid_acc_list.append(valid_acc)
        train_loss_list.append(train_loss), valid_loss_list.append(valid_loss)
        print(f" EPOCH {epoch_idx}. Progress: {epoch_idx/num_epoch*100}%. ")
        print(f" Train accuracy: {train_acc}. Valid accuracy: {valid_acc}")
        print(f" Train CE loss: {train_loss}. Valid CE loss: {valid_loss}")

        if valid_acc > base_valid_acc:
```

```
            base_valid_acc = valid_acc
            torch.save(model.state_dict(), os.path.join(path_model,␣
↪'model_shallowCNN_bs_{}_lr_{}_epoch_{}.pth'.format(str(bs), str(lr),␣
↪str(num_epoch))))
```

```
Training input shape torch.Size([1692, 22, 1000])
Validation input shape torch.Size([423, 22, 1000])
```

```
/tmp/ipykernel_1222/3743213622.py:49: UserWarning: Implicit dimension choice for
softmax has been deprecated. Change the call to include dim=X as an argument.
  pred = softmax(self.ln1(x_embd))
```

```
 EPOCH 0. Progress: 0.0%.
 Train accuracy: 0.25591015815734863. Valid accuracy: 0.2600472867488861
 Train CE loss: 0.3465652133274304. Valid CE loss: 0.34738125896904765
 EPOCH 10. Progress: 1.0%.
 Train accuracy: 0.25591015815734863. Valid accuracy: 0.2600472867488861
 Train CE loss: 0.3465599419095556. Valid CE loss: 0.347365885479794
 EPOCH 20. Progress: 2.0%.
 Train accuracy: 0.256501168012619. Valid accuracy: 0.2600472867488861
 Train CE loss: 0.34643832060462193. Valid CE loss: 0.347255830787316
 EPOCH 30. Progress: 3.0%.
 Train accuracy: 0.3705673813819885. Valid accuracy: 0.3356974124908447
 Train CE loss: 0.33522355577624435. Valid CE loss: 0.3403340093351143
 EPOCH 40. Progress: 4.0%.
 Train accuracy: 0.40070921182632446. Valid accuracy: 0.368794322013855
 Train CE loss: 0.32534531998859795. Valid CE loss: 0.3360229293787169
 EPOCH 50. Progress: 5.0%.
 Train accuracy: 0.45035460591316223. Valid accuracy: 0.38770684599876404
 Train CE loss: 0.3191464067773616. Valid CE loss: 0.3325724011335531
 EPOCH 60. Progress: 6.0%.
 Train accuracy: 0.4403073191642761. Valid accuracy: 0.3995271921157837
 Train CE loss: 0.3181444317031978. Valid CE loss: 0.33129514320522335
 EPOCH 70. Progress: 7.000000000000001%.
 Train accuracy: 0.4858156144618988. Valid accuracy: 0.41134750843048096
 Train CE loss: 0.3120472551660335. Valid CE loss: 0.32764245897320143
 EPOCH 80. Progress: 8.0%.
 Train accuracy: 0.5159574747085571. Valid accuracy: 0.4160756468772888
 Train CE loss: 0.30707724786960205. Valid CE loss: 0.3245218012913463
 EPOCH 90. Progress: 9.0%.
 Train accuracy: 0.5395981073379517. Valid accuracy: 0.4160756468772888
 Train CE loss: 0.303457887354472. Valid CE loss: 0.3225215982038079
 EPOCH 100. Progress: 10.0%.
 Train accuracy: 0.5585106611251831. Valid accuracy: 0.4137115776538849
 Train CE loss: 0.2995318135612118. Valid CE loss: 0.3209464968519008
 EPOCH 110. Progress: 11.0%.
 Train accuracy: 0.563829779624939. Valid accuracy: 0.44208037853240967
 Train CE loss: 0.2981065193952962. Valid CE loss: 0.31802469534231415
```

```
EPOCH 120. Progress: 12.0%.
Train accuracy: 0.5963357090950012. Valid accuracy: 0.44208037853240967
Train CE loss: 0.2908208518515806. Valid CE loss: 0.3167192563942984
EPOCH 130. Progress: 13.0%.
Train accuracy: 0.6247044801712036. Valid accuracy: 0.4728132486343384
Train CE loss: 0.28533038644925923. Valid CE loss: 0.3130400221680355
EPOCH 140. Progress: 14.000000000000002%.
Train accuracy: 0.5721040368080139. Valid accuracy: 0.4680851101875305
Train CE loss: 0.29118366363341647. Valid CE loss: 0.3125943413299308
EPOCH 150. Progress: 15.0%.
Train accuracy: 0.6270685791969299. Valid accuracy: 0.4728132486343384
Train CE loss: 0.2813230064250617. Valid CE loss: 0.31086270338536437
EPOCH 160. Progress: 16.0%.
Train accuracy: 0.6589834690093994. Valid accuracy: 0.4893617033958435
Train CE loss: 0.2759812071250122. Valid CE loss: 0.30861511622196675
EPOCH 170. Progress: 17.0%.
Train accuracy: 0.6678487062454224. Valid accuracy: 0.5059101581573486
Train CE loss: 0.2740505235555888. Valid CE loss: 0.30597928469344515
EPOCH 180. Progress: 18.0%.
Train accuracy: 0.6317967176437378. Valid accuracy: 0.4893617033958435
Train CE loss: 0.27691933826210935. Valid CE loss: 0.31018901702641877
EPOCH 190. Progress: 19.0%.
Train accuracy: 0.7027186751365662. Valid accuracy: 0.5106382966041565
Train CE loss: 0.2650116086710711. Valid CE loss: 0.30300060922090605
EPOCH 200. Progress: 20.0%.
Train accuracy: 0.7139480113983154. Valid accuracy: 0.5177304744720459
Train CE loss: 0.2626461531112662. Valid CE loss: 0.30171712634129444
EPOCH 210. Progress: 21.0%.
Train accuracy: 0.7281323671340942. Valid accuracy: 0.5295508503913879
Train CE loss: 0.2602756080122986. Valid CE loss: 0.30110329784118256
EPOCH 220. Progress: 22.0%.
Train accuracy: 0.7364066243171692. Valid accuracy: 0.5248227119445801
Train CE loss: 0.257524453121156. Valid CE loss: 0.3005332593094936
EPOCH 230. Progress: 23.0%.
Train accuracy: 0.7429078221321106. Valid accuracy: 0.5248227119445801
Train CE loss: 0.25509153257165956. Valid CE loss: 0.29889662454596083
EPOCH 240. Progress: 24.0%.
Train accuracy: 0.7618203163146973. Valid accuracy: 0.5319148898124695
Train CE loss: 0.25201310340676186. Valid CE loss: 0.29928419000995354
EPOCH 250. Progress: 25.0%.
Train accuracy: 0.7665484547615051. Valid accuracy: 0.5366430282592773
Train CE loss: 0.2496520101883169. Valid CE loss: 0.2988627133076354
EPOCH 260. Progress: 26.0%.
Train accuracy: 0.7677304744720459. Valid accuracy: 0.5484633445739746
Train CE loss: 0.2488061875175359. Valid CE loss: 0.2966887907215326
EPOCH 270. Progress: 27.0%.
Train accuracy: 0.7801418304443359. Valid accuracy: 0.5295508503913879
Train CE loss: 0.2470721311814396. Valid CE loss: 0.2991078291660787
```

```
EPOCH 280. Progress: 28.000000000000004%.
Train accuracy: 0.7813238501548767. Valid accuracy: 0.5319148898124695
Train CE loss: 0.24513923952765498. Valid CE loss: 0.29809806448348025
EPOCH 290. Progress: 28.999999999999996%.
Train accuracy: 0.8002364039421082. Valid accuracy: 0.5437352061271667
Train CE loss: 0.24150511840838348. Valid CE loss: 0.2978263693216563
EPOCH 300. Progress: 30.0%.
Train accuracy: 0.7996453642845154. Valid accuracy: 0.5602836608886719
Train CE loss: 0.24136155984644067. Valid CE loss: 0.2961642326474472
EPOCH 310. Progress: 31.0%.
Train accuracy: 0.813238799571991. Valid accuracy: 0.5531914830207825
Train CE loss: 0.23796468610318278. Valid CE loss: 0.29546456119974734
EPOCH 320. Progress: 32.0%.
Train accuracy: 0.8102836608886719. Valid accuracy: 0.567375898361206
Train CE loss: 0.23799170990503143. Valid CE loss: 0.294286533027676
EPOCH 330. Progress: 33.0%.
Train accuracy: 0.8209219574928284. Valid accuracy: 0.5531914830207825
Train CE loss: 0.2361806594099559. Valid CE loss: 0.29559250827658545
EPOCH 340. Progress: 34.0%.
Train accuracy: 0.8238770961761475. Valid accuracy: 0.5508274435997009
Train CE loss: 0.2356053971934262. Valid CE loss: 0.2944891771244383
EPOCH 350. Progress: 35.0%.
Train accuracy: 0.8167848587036133. Valid accuracy: 0.5531914830207825
Train CE loss: 0.23703259486819553. Valid CE loss: 0.29613468621639494
EPOCH 360. Progress: 36.0%.
Train accuracy: 0.8327423334121704. Valid accuracy: 0.5744680762290955
Train CE loss: 0.23327326908461027. Valid CE loss: 0.2924906763624638
EPOCH 370. Progress: 37.0%.
Train accuracy: 0.8404255509376526. Valid accuracy: 0.5697399377822876
Train CE loss: 0.23013619634699314. Valid CE loss: 0.29388034132355495
EPOCH 380. Progress: 38.0%.
Train accuracy: 0.8374704718589783. Valid accuracy: 0.5484633445739746
Train CE loss: 0.23146311392739027. Valid CE loss: 0.2954922739777441
EPOCH 390. Progress: 39.0%.
Train accuracy: 0.8481087684631348. Valid accuracy: 0.5768321752548218
Train CE loss: 0.22877408468976934. Valid CE loss: 0.2915103730016848
EPOCH 400. Progress: 40.0%.
Train accuracy: 0.8498817682266235. Valid accuracy: 0.5768321752548218
Train CE loss: 0.22676298078633933. Valid CE loss: 0.2930841861605362
EPOCH 410. Progress: 41.0%.
Train accuracy: 0.8486997485160828. Valid accuracy: 0.5626477599143982
Train CE loss: 0.22745559405068697. Valid CE loss: 0.2941720997188108
EPOCH 420. Progress: 42.0%.
Train accuracy: 0.8534278869628906. Valid accuracy: 0.5815603137016296
Train CE loss: 0.2251569463851604. Valid CE loss: 0.292191405104689
EPOCH 430. Progress: 43.0%.
Train accuracy: 0.8528369069099426. Valid accuracy: 0.567375898361206
Train CE loss: 0.22593877755158337. Valid CE loss: 0.29257626806872394
```

```
EPOCH 440. Progress: 44.0%.
Train accuracy: 0.8581560254096985. Valid accuracy: 0.5839243531227112
Train CE loss: 0.22444917096032035. Valid CE loss: 0.2917566513620652
EPOCH 450. Progress: 45.0%.
Train accuracy: 0.8416075706481934. Valid accuracy: 0.5555555820465088
Train CE loss: 0.22916862269947152. Valid CE loss: 0.29293603818185504
EPOCH 460. Progress: 46.0%.
Train accuracy: 0.8617021441459656. Valid accuracy: 0.5839243531227112
Train CE loss: 0.22305821498798703. Valid CE loss: 0.291736580237711
EPOCH 470. Progress: 47.0%.
Train accuracy: 0.8617021441459656. Valid accuracy: 0.5744680762290955
Train CE loss: 0.2225184076362186. Valid CE loss: 0.29211103000257593
EPOCH 480. Progress: 48.0%.
Train accuracy: 0.8634752035140991. Valid accuracy: 0.5626477599143982
Train CE loss: 0.22129177395474545. Valid CE loss: 0.29309532081545386
EPOCH 490. Progress: 49.0%.
Train accuracy: 0.8646572232246399. Valid accuracy: 0.5626477599143982
Train CE loss: 0.22140493118875698. Valid CE loss: 0.2943045088303568
EPOCH 500. Progress: 50.0%.
Train accuracy: 0.8540189266204834. Valid accuracy: 0.5579196214675903
Train CE loss: 0.22649762872826687. Valid CE loss: 0.2952215034629154
EPOCH 510. Progress: 51.0%.
Train accuracy: 0.8682032823562622. Valid accuracy: 0.5697399377822876
Train CE loss: 0.2195659512675964. Valid CE loss: 0.29287935092375916
EPOCH 520. Progress: 52.0%.
Train accuracy: 0.8693853616714478. Valid accuracy: 0.5744680762290955
Train CE loss: 0.21915991537396226. Valid CE loss: 0.2925944708763285
EPOCH 530. Progress: 53.0%.
Train accuracy: 0.8699763417243958. Valid accuracy: 0.5650117993354797
Train CE loss: 0.21922141567189643. Valid CE loss: 0.2941083903853775
EPOCH 540. Progress: 54.0%.
Train accuracy: 0.8711584210395813. Valid accuracy: 0.5579196214675903
Train CE loss: 0.21940111419974206. Valid CE loss: 0.2933277046708632
EPOCH 550. Progress: 55.00000000000001%.
Train accuracy: 0.8717494010925293. Valid accuracy: 0.5768321752548218
Train CE loss: 0.21809492550843151. Valid CE loss: 0.29237984173686793
EPOCH 560. Progress: 56.00000000000001%.
Train accuracy: 0.8723404407501221. Valid accuracy: 0.5744680762290955
Train CE loss: 0.2177905424125933. Valid CE loss: 0.29271967163040846
EPOCH 570. Progress: 56.99999999999999%.
Train accuracy: 0.8735224604606628. Valid accuracy: 0.5839243531227112
Train CE loss: 0.21786639908104077. Valid CE loss: 0.2917687791459104
EPOCH 580. Progress: 57.99999999999999%.
Train accuracy: 0.8741135001182556. Valid accuracy: 0.5460993051528931
Train CE loss: 0.21780740517251035. Valid CE loss: 0.29601176593884226
EPOCH 590. Progress: 59.0%.
Train accuracy: 0.8758864998817444. Valid accuracy: 0.5744680762290955
Train CE loss: 0.21711388670665435. Valid CE loss: 0.2908125474661518
```

```
EPOCH 600. Progress: 60.0%.
Train accuracy: 0.8776595592498779. Valid accuracy: 0.5697399377822876
Train CE loss: 0.21705853872124467. Valid CE loss: 0.2922792081009975
EPOCH 610. Progress: 61.0%.
Train accuracy: 0.8794326186180115. Valid accuracy: 0.5815603137016296
Train CE loss: 0.21608966433973742. Valid CE loss: 0.29061257134656254
EPOCH 620. Progress: 62.0%.
Train accuracy: 0.8800236582756042. Valid accuracy: 0.5697399377822876
Train CE loss: 0.21580384555437887. Valid CE loss: 0.29225298022547513
EPOCH 630. Progress: 63.0%.
Train accuracy: 0.8806146383285522. Valid accuracy: 0.588652491569519
Train CE loss: 0.21555604012440846. Valid CE loss: 0.2897963955047283
EPOCH 640. Progress: 64.0%.
Train accuracy: 0.881205677986145. Valid accuracy: 0.5768321752548218
Train CE loss: 0.21550299815534135. Valid CE loss: 0.29144150934602636
EPOCH 650. Progress: 65.0%.
Train accuracy: 0.881205677986145. Valid accuracy: 0.5839243531227112
Train CE loss: 0.2159918012278018. Valid CE loss: 0.29066266320275924
EPOCH 660. Progress: 66.0%.
Train accuracy: 0.8829787373542786. Valid accuracy: 0.5650117993354797
Train CE loss: 0.2151932483164695. Valid CE loss: 0.2931223445468479
EPOCH 670. Progress: 67.0%.
Train accuracy: 0.8835697174072266. Valid accuracy: 0.5791962146759033
Train CE loss: 0.21468998960287577. Valid CE loss: 0.29172571345142156
EPOCH 680. Progress: 68.0%.
Train accuracy: 0.8841607570648193. Valid accuracy: 0.5697399377822876
Train CE loss: 0.21546138717350385. Valid CE loss: 0.29368445988806147
EPOCH 690. Progress: 69.0%.
Train accuracy: 0.8841607570648193. Valid accuracy: 0.5697399377822876
Train CE loss: 0.2143544747403891. Valid CE loss: 0.29275279321287256
EPOCH 700. Progress: 70.0%.
Train accuracy: 0.8841607570648193. Valid accuracy: 0.5768321752548218
Train CE loss: 0.2145930081253638. Valid CE loss: 0.2911656215681252
EPOCH 710. Progress: 71.0%.
Train accuracy: 0.8841607570648193. Valid accuracy: 0.5531914830207825
Train CE loss: 0.21458037077816947. Valid CE loss: 0.2942156646550406
EPOCH 720. Progress: 72.0%.
Train accuracy: 0.8847517967224121. Valid accuracy: 0.567375898361206
Train CE loss: 0.21440227579845042. Valid CE loss: 0.2921193001118112
EPOCH 730. Progress: 73.0%.
Train accuracy: 0.8859338164329529. Valid accuracy: 0.5744680762290955
Train CE loss: 0.21387767118906018. Valid CE loss: 0.2917753973751203
EPOCH 740. Progress: 74.0%.
Train accuracy: 0.8859338164329529. Valid accuracy: 0.5768321752548218
Train CE loss: 0.21421186710503085. Valid CE loss: 0.29138329885811953
EPOCH 750. Progress: 75.0%.
Train accuracy: 0.8865247964859009. Valid accuracy: 0.5791962146759033
Train CE loss: 0.2138620476914354. Valid CE loss: 0.29199667771657306
```

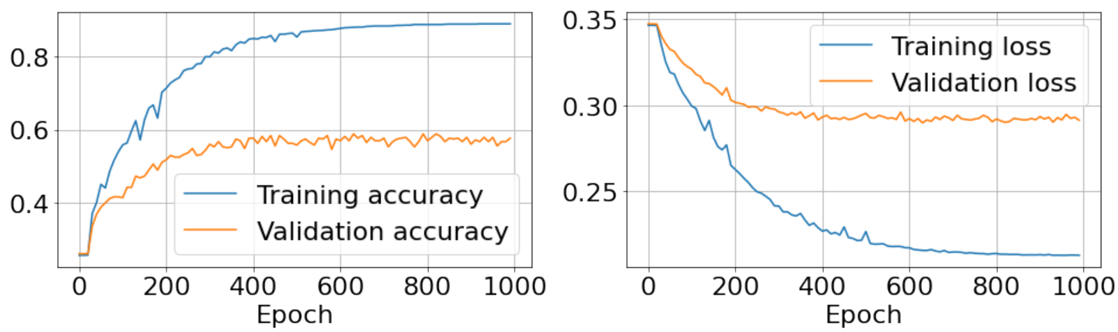
```
EPOCH 760. Progress: 76.0%.
Train accuracy: 0.8865247964859009. Valid accuracy: 0.567375898361206
Train CE loss: 0.21368597229462707. Valid CE loss: 0.29245453058405124
EPOCH 770. Progress: 77.0%.
Train accuracy: 0.8882978558540344. Valid accuracy: 0.5531914830207825
Train CE loss: 0.21368510852046046. Valid CE loss: 0.29418014099130113
EPOCH 780. Progress: 78.0%.
Train accuracy: 0.8882978558540344. Valid accuracy: 0.567375898361206
Train CE loss: 0.2132912363425496. Valid CE loss: 0.29299471409326466
EPOCH 790. Progress: 79.0%.
Train accuracy: 0.8882978558540344. Valid accuracy: 0.588652491569519
Train CE loss: 0.21361399989742477. Valid CE loss: 0.2904347292638558
EPOCH 800. Progress: 80.0%.
Train accuracy: 0.8882978558540344. Valid accuracy: 0.5697399377822876
Train CE loss: 0.21370358493874822. Valid CE loss: 0.29203113751490345
EPOCH 810. Progress: 81.0%.
Train accuracy: 0.8882978558540344. Valid accuracy: 0.5791962146759033
Train CE loss: 0.21329976054652644. Valid CE loss: 0.2909148756775732
EPOCH 820. Progress: 82.0%.
Train accuracy: 0.8882978558540344. Valid accuracy: 0.588652491569519
Train CE loss: 0.21332240488653206. Valid CE loss: 0.2900556142730352
EPOCH 830. Progress: 83.0%.
Train accuracy: 0.8882978558540344. Valid accuracy: 0.5815603137016296
Train CE loss: 0.21318204834810667. Valid CE loss: 0.29056966389324645
EPOCH 840. Progress: 84.0%.
Train accuracy: 0.8888888955116272. Valid accuracy: 0.567375898361206
Train CE loss: 0.21318682524470292. Valid CE loss: 0.29237428497760853
EPOCH 850. Progress: 85.0%.
Train accuracy: 0.88947993516922. Valid accuracy: 0.5768321752548218
Train CE loss: 0.21316918783576777. Valid CE loss: 0.2919496699427882
EPOCH 860. Progress: 86.0%.
Train accuracy: 0.88947993516922. Valid accuracy: 0.5744680762290955
Train CE loss: 0.21288529369566175. Valid CE loss: 0.29151183324502716
EPOCH 870. Progress: 87.0%.
Train accuracy: 0.88947993516922. Valid accuracy: 0.5768321752548218
Train CE loss: 0.21293719112873077. Valid CE loss: 0.29150672147741835
EPOCH 880. Progress: 88.0%.
Train accuracy: 0.88947993516922. Valid accuracy: 0.567375898361206
Train CE loss: 0.21295745857500298. Valid CE loss: 0.29230808591729923
EPOCH 890. Progress: 89.0%.
Train accuracy: 0.88947993516922. Valid accuracy: 0.5791962146759033
Train CE loss: 0.21291359547463998. Valid CE loss: 0.2915834464643582
EPOCH 900. Progress: 90.0%.
Train accuracy: 0.88947993516922. Valid accuracy: 0.5602836608886719
Train CE loss: 0.21303559085860602. Valid CE loss: 0.2934028189233009
EPOCH 910. Progress: 91.0%.
Train accuracy: 0.88947993516922. Valid accuracy: 0.5721040368080139
Train CE loss: 0.21283016919244266. Valid CE loss: 0.2920232329244591
```

```
EPOCH 920. Progress: 92.0%.
Train accuracy: 0.890070915222168. Valid accuracy: 0.567375898361206
Train CE loss: 0.21312596080010102. Valid CE loss: 0.2929916245153892
EPOCH 930. Progress: 93.0%.
Train accuracy: 0.890070915222168. Valid accuracy: 0.5791962146759033
Train CE loss: 0.21270892443245465. Valid CE loss: 0.29022741374112754
EPOCH 940. Progress: 94.0%.
Train accuracy: 0.890070915222168. Valid accuracy: 0.567375898361206
Train CE loss: 0.2127728832810765. Valid CE loss: 0.2928040350300764
EPOCH 950. Progress: 95.0%.
Train accuracy: 0.890070915222168. Valid accuracy: 0.5791962146759033
Train CE loss: 0.21268857684400347. Valid CE loss: 0.29148765996838294
EPOCH 960. Progress: 96.0%.
Train accuracy: 0.890070915222168. Valid accuracy: 0.5555555820465088
Train CE loss: 0.21272508003892066. Valid CE loss: 0.29469213164444513
EPOCH 970. Progress: 97.0%.
Train accuracy: 0.890070915222168. Valid accuracy: 0.567375898361206
Train CE loss: 0.2128465439129101. Valid CE loss: 0.2924029648163076
EPOCH 980. Progress: 98.0%.
Train accuracy: 0.890070915222168. Valid accuracy: 0.567375898361206
Train CE loss: 0.21281333602348398. Valid CE loss: 0.29296693452424755
EPOCH 990. Progress: 99.0%.
Train accuracy: 0.890070915222168. Valid accuracy: 0.5768321752548218
Train CE loss: 0.21270742508677445. Valid CE loss: 0.29127141248531657
```

[ ]:
```
np.save(os.path.join(path_model, 'train_acc_list.npy'), np.asarray(torch.
 ↪Tensor(train_acc_list).cpu()))
np.save(os.path.join(path_model, 'train_loss_list.npy'), np.asarray(torch.
 ↪Tensor(train_loss_list).cpu()))
np.save(os.path.join(path_model, 'valid_acc_list.npy'), np.asarray(torch.
 ↪Tensor(valid_acc_list).cpu()))
np.save(os.path.join(path_model, 'valid_loss_list.npy'), np.asarray(torch.
 ↪Tensor(valid_loss_list).cpu()))
```

[ ]:
```
def func_plotCNNStats(path_figures, train_acc_list, train_loss_list,␣
 ↪valid_acc_list, valid_loss_list):
    fig, axes = plt.subplots(1, 2)
    fig.set_size_inches(8*2,4)
    epoch_grid = np.arange(0, 1000, 10)
    axes[0].plot(epoch_grid, train_acc_list, label='Training accuracy')
    axes[0].plot(epoch_grid, valid_acc_list, label='Validation accuracy')
    axes[0].set_xlabel('Epoch')
    axes[0].grid(True)
    axes[0].legend()
    # axes[0].title.set_text('Train and validation accuracy')

    axes[1].plot(epoch_grid, train_loss_list, label='Training loss')
```

```
    axes[1].plot(epoch_grid, valid_loss_list, label='Validation loss')
    axes[1].set_xlabel('Epoch')
    axes[1].grid(True)
    axes[1].legend()
    # axes[1].title.set_text('Train and validation cross entropy loss')

    fig.savefig(os.path.join(path_figures, 'training_stats.pdf'),␣
 ↪bbox_inches='tight')

train_acc_list, train_loss_list = np.load(os.path.join(path_model,␣
 ↪'train_acc_list.npy')), np.load(os.path.join(path_model, 'train_loss_list.
 ↪npy'))
valid_acc_list, valid_loss_list = np.load(os.path.join(path_model,␣
 ↪'valid_acc_list.npy')), np.load(os.path.join(path_model, 'valid_loss_list.
 ↪npy'))

func_plotCNNStats(path_figures, train_acc_list, train_loss_list,␣
 ↪valid_acc_list, valid_loss_list)
```



[ ]:
```
# start testing
model_name = 'model_shallowCNN_bs_{}_lr_{}_epoch_{}.pth'.format(str(bs),␣
 ↪str(lr), str(num_epoch))
model_path = os.path.join(path_model, model_name)

X_test = torch.Tensor(X_test)
y_test = torch.Tensor(y_test)

EEG_test_dataset = DataLoader(TensorDataset(X_test, y_test))

model = ShallowCNN()
model.load_state_dict(torch.load(model_path))
model = model.cuda()
model.eval()
```

```
test_numOfCorrectLabels = 0
test_embed_list = []
test_pred_list = []
for test_data, test_label in EEG_test_dataset:
    # test_data.to('cuda')

    test_input = torch.reshape(test_data, (test_data.shape[0], 1, test_data.
 ↪shape[1], test_data.shape[2])) # expand one dimension for the channel for␣
 ↪the EGG
    test_input = test_input.to('cuda')

    test_pred_onehot, test_x_embd = model(test_input)
    test_embed_list.append(test_x_embd.detach().cpu().numpy())

    test_pred = torch.argmax(test_pred_onehot, dim=1).cpu()
    test_pred_list.append(test_pred.cpu().numpy())

    test_numOfCorrectLabels += (test_pred == test_label).float().sum()

test_accuracy = test_numOfCorrectLabels / len(EEG_test_dataset.dataset)

print('Test accuracy of ShallowCNN is', test_accuracy.numpy())
```

```
/tmp/ipykernel_1222/3743213622.py:49: UserWarning: Implicit dimension choice for
softmax has been deprecated. Change the call to include dim=X as an argument.
  pred = softmax(self.ln1(x_embd))
```

```
Test accuracy of ShallowCNN is 0.5756208
```

[ ]:
```
test_shallowCNN_embed = np.vstack(test_embed_list)
print(test_shallowCNN_embed.shape)
pred_shallowCNN = np.vstack(test_pred_list)
print(pred_shallowCNN.shape)
```

```
(443, 2440)
(443, 1)
```

[ ]:
```
import matplotlib.pyplot as plt

trial_index_list = [1, 100, 200, 300, 400]

fig, axes = plt.subplots(5, 6)
fig.set_size_inches((24, 20))

perplexity = 30


for fig_idx, idx in enumerate(trial_index_list):
```

```
    # wavelet db1 tsne
    db1_tsne = TSNE(perplexity=perplexity).
↪fit_transform(func_testReshape(test_cA_db1))
    scatter = axes[fig_idx][0].scatter(db1_tsne[:,0], db1_tsne[:,1], c=pred_db1)
    legend = axes[fig_idx][0].legend(*scatter.legend_elements(), prop={'size':
↪6}, loc='upper left')
    axes[fig_idx][0].add_artist(legend)
    axes[fig_idx][0].set_xticklabels([])
    axes[fig_idx][0].set_yticklabels([])
    # axes[fig_idx][0].axis('off')

    db2_tsne = TSNE(perplexity=perplexity).
↪fit_transform(func_testReshape(test_cA_db2))
    scatter = axes[fig_idx][1].scatter(db2_tsne[:,0], db2_tsne[:,1], c=pred_db2)
    legend = axes[fig_idx][1].legend(*scatter.legend_elements(), prop={'size':
↪6}, loc='upper left')
    axes[fig_idx][1].add_artist(legend)
    axes[fig_idx][1].set_xticklabels([])
    axes[fig_idx][1].set_yticklabels([])
    # axes[fig_idx][1].axis('off')

    db3_tsne = TSNE(perplexity=perplexity).
↪fit_transform(func_testReshape(test_cA_db3))
    scatter = axes[fig_idx][2].scatter(db3_tsne[:,0], db3_tsne[:,1], c=pred_db3)
    legend = axes[fig_idx][2].legend(*scatter.legend_elements(), prop={'size':
↪6}, loc='upper left')
    axes[fig_idx][2].add_artist(legend)
    axes[fig_idx][2].set_xticklabels([])
    axes[fig_idx][2].set_yticklabels([])
    # axes[fig_idx][2].axis('off')

    db4_tsne = TSNE(perplexity=perplexity).
↪fit_transform(func_testReshape(test_cA_db4))
    scatter = axes[fig_idx][3].scatter(db4_tsne[:,0], db4_tsne[:,1], c=pred_db4)
    legend = axes[fig_idx][3].legend(*scatter.legend_elements(), prop={'size':
↪6}, loc='upper left')
    axes[fig_idx][3].add_artist(legend)
    axes[fig_idx][3].set_xticklabels([])
    axes[fig_idx][3].set_yticklabels([])
    # axes[fig_idx][3].axis('off')

    tsne_2d = TSNE(perplexity=perplexity).fit_transform(tsne_test)
    scatter = axes[fig_idx][4].scatter(tsne_2d[:,0], tsne_2d[:,1], c=pred_tsne)
    legend = axes[fig_idx][4].legend(*scatter.legend_elements(), prop={'size':
↪6}, loc='upper left')
    axes[fig_idx][4].add_artist(legend)
```

```
    axes[fig_idx][4].set_xticklabels([])
    axes[fig_idx][4].set_yticklabels([])
    # axes[fig_idx][4].axis('off')

    CNN_tsne = TSNE(perplexity=perplexity).fit_transform(test_shallowCNN_embed)
    scatter = axes[fig_idx][5].scatter(CNN_tsne[:,0], CNN_tsne[:,1],␣
↪c=pred_shallowCNN)
    legend = axes[fig_idx][5].legend(*scatter.legend_elements(), prop={'size':
↪6}, loc='upper left')
    axes[fig_idx][5].add_artist(legend)
    axes[fig_idx][5].set_xticklabels([])
    axes[fig_idx][5].set_yticklabels([])
    # axes[fig_idx][5].axis('off')

    axes[fig_idx][0].set_ylabel('Trial {}'.format(trial_index_list[fig_idx]))
    # break

axes[fig_idx][0].set_xlabel('Wavelet db1+SVM')
axes[fig_idx][1].set_xlabel('Wavelet db2+SVM')
axes[fig_idx][2].set_xlabel('Wavelet db3+SVM')
axes[fig_idx][3].set_xlabel('Wavelet db4+SVM')
axes[fig_idx][4].set_xlabel('t-SNE+SVM')
axes[fig_idx][5].set_xlabel('Shallow CNN')

fig.savefig(os.path.join(path_figures, 'embedding_cmp.pdf'),␣
↪bbox_inches='tight')
```

```
/home/xiaoranzhang/anaconda3/envs/torch_env/lib/python3.9/site-
packages/sklearn/manifold/_t_sne.py:780: FutureWarning: The default
initialization in TSNE will change from 'random' to 'pca' in 1.2.
  warnings.warn(
/home/xiaoranzhang/anaconda3/envs/torch_env/lib/python3.9/site-
packages/sklearn/manifold/_t_sne.py:790: FutureWarning: The default learning
rate in TSNE will change from 200.0 to 'auto' in 1.2.
  warnings.warn(
/home/xiaoranzhang/anaconda3/envs/torch_env/lib/python3.9/site-
packages/sklearn/manifold/_t_sne.py:780: FutureWarning: The default
initialization in TSNE will change from 'random' to 'pca' in 1.2.
  warnings.warn(
/home/xiaoranzhang/anaconda3/envs/torch_env/lib/python3.9/site-
packages/sklearn/manifold/_t_sne.py:790: FutureWarning: The default learning
rate in TSNE will change from 200.0 to 'auto' in 1.2.
  warnings.warn(
/home/xiaoranzhang/anaconda3/envs/torch_env/lib/python3.9/site-
packages/sklearn/manifold/_t_sne.py:780: FutureWarning: The default
initialization in TSNE will change from 'random' to 'pca' in 1.2.
  warnings.warn(
```

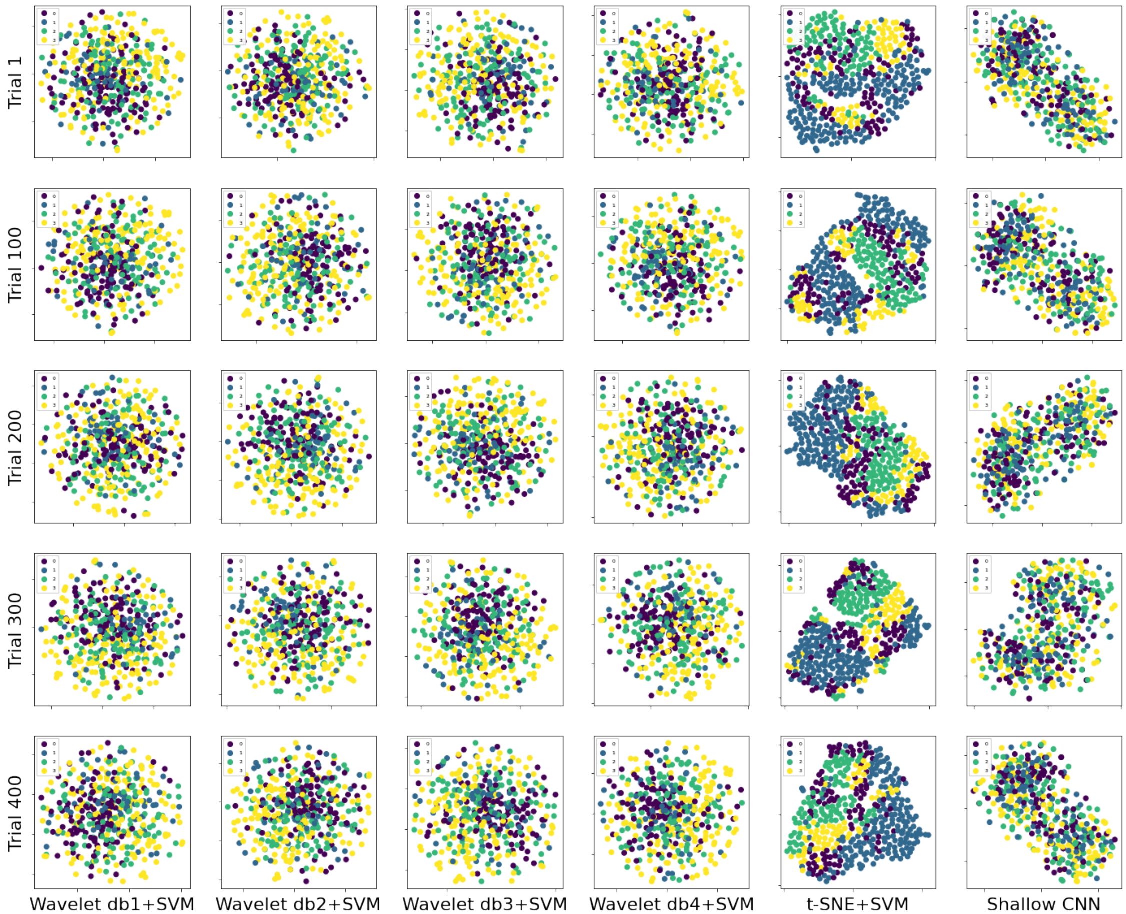
```
/home/xiaoranzhang/anaconda3/envs/torch_env/lib/python3.9/site-
packages/sklearn/manifold/_t_sne.py:790: FutureWarning: The default learning
rate in TSNE will change from 200.0 to 'auto' in 1.2.
  warnings.warn(
/home/xiaoranzhang/anaconda3/envs/torch_env/lib/python3.9/site-
packages/sklearn/manifold/_t_sne.py:780: FutureWarning: The default
initialization in TSNE will change from 'random' to 'pca' in 1.2.
  warnings.warn(
/home/xiaoranzhang/anaconda3/envs/torch_env/lib/python3.9/site-
packages/sklearn/manifold/_t_sne.py:790: FutureWarning: The default learning
rate in TSNE will change from 200.0 to 'auto' in 1.2.
  warnings.warn(
/home/xiaoranzhang/anaconda3/envs/torch_env/lib/python3.9/site-
packages/sklearn/manifold/_t_sne.py:780: FutureWarning: The default
initialization in TSNE will change from 'random' to 'pca' in 1.2.
  warnings.warn(
/home/xiaoranzhang/anaconda3/envs/torch_env/lib/python3.9/site-
packages/sklearn/manifold/_t_sne.py:790: FutureWarning: The default learning
rate in TSNE will change from 200.0 to 'auto' in 1.2.
  warnings.warn(
/home/xiaoranzhang/anaconda3/envs/torch_env/lib/python3.9/site-
packages/sklearn/manifold/_t_sne.py:780: FutureWarning: The default
initialization in TSNE will change from 'random' to 'pca' in 1.2.
  warnings.warn(
/home/xiaoranzhang/anaconda3/envs/torch_env/lib/python3.9/site-
packages/sklearn/manifold/_t_sne.py:790: FutureWarning: The default learning
rate in TSNE will change from 200.0 to 'auto' in 1.2.
  warnings.warn(
/home/xiaoranzhang/anaconda3/envs/torch_env/lib/python3.9/site-
packages/sklearn/manifold/_t_sne.py:780: FutureWarning: The default
initialization in TSNE will change from 'random' to 'pca' in 1.2.
  warnings.warn(
/home/xiaoranzhang/anaconda3/envs/torch_env/lib/python3.9/site-
packages/sklearn/manifold/_t_sne.py:790: FutureWarning: The default learning
rate in TSNE will change from 200.0 to 'auto' in 1.2.
  warnings.warn(
/home/xiaoranzhang/anaconda3/envs/torch_env/lib/python3.9/site-
packages/sklearn/manifold/_t_sne.py:780: FutureWarning: The default
initialization in TSNE will change from 'random' to 'pca' in 1.2.
  warnings.warn(
/home/xiaoranzhang/anaconda3/envs/torch_env/lib/python3.9/site-
packages/sklearn/manifold/_t_sne.py:790: FutureWarning: The default learning
rate in TSNE will change from 200.0 to 'auto' in 1.2.
  warnings.warn(
/home/xiaoranzhang/anaconda3/envs/torch_env/lib/python3.9/site-
packages/sklearn/manifold/_t_sne.py:780: FutureWarning: The default
initialization in TSNE will change from 'random' to 'pca' in 1.2.
  warnings.warn(
```

```
/home/xiaoranzhang/anaconda3/envs/torch_env/lib/python3.9/site-
packages/sklearn/manifold/_t_sne.py:790: FutureWarning: The default learning
rate in TSNE will change from 200.0 to 'auto' in 1.2.
  warnings.warn(
/home/xiaoranzhang/anaconda3/envs/torch_env/lib/python3.9/site-
packages/sklearn/manifold/_t_sne.py:780: FutureWarning: The default
initialization in TSNE will change from 'random' to 'pca' in 1.2.
  warnings.warn(
/home/xiaoranzhang/anaconda3/envs/torch_env/lib/python3.9/site-
packages/sklearn/manifold/_t_sne.py:790: FutureWarning: The default learning
rate in TSNE will change from 200.0 to 'auto' in 1.2.
  warnings.warn(
/home/xiaoranzhang/anaconda3/envs/torch_env/lib/python3.9/site-
packages/sklearn/manifold/_t_sne.py:780: FutureWarning: The default
initialization in TSNE will change from 'random' to 'pca' in 1.2.
  warnings.warn(
/home/xiaoranzhang/anaconda3/envs/torch_env/lib/python3.9/site-
packages/sklearn/manifold/_t_sne.py:790: FutureWarning: The default learning
rate in TSNE will change from 200.0 to 'auto' in 1.2.
  warnings.warn(
/home/xiaoranzhang/anaconda3/envs/torch_env/lib/python3.9/site-
packages/sklearn/manifold/_t_sne.py:780: FutureWarning: The default
initialization in TSNE will change from 'random' to 'pca' in 1.2.
  warnings.warn(
/home/xiaoranzhang/anaconda3/envs/torch_env/lib/python3.9/site-
packages/sklearn/manifold/_t_sne.py:790: FutureWarning: The default learning
rate in TSNE will change from 200.0 to 'auto' in 1.2.
  warnings.warn(
/home/xiaoranzhang/anaconda3/envs/torch_env/lib/python3.9/site-
packages/sklearn/manifold/_t_sne.py:780: FutureWarning: The default
initialization in TSNE will change from 'random' to 'pca' in 1.2.
  warnings.warn(
/home/xiaoranzhang/anaconda3/envs/torch_env/lib/python3.9/site-
packages/sklearn/manifold/_t_sne.py:790: FutureWarning: The default learning
rate in TSNE will change from 200.0 to 'auto' in 1.2.
  warnings.warn(
/home/xiaoranzhang/anaconda3/envs/torch_env/lib/python3.9/site-
packages/sklearn/manifold/_t_sne.py:780: FutureWarning: The default
initialization in TSNE will change from 'random' to 'pca' in 1.2.
  warnings.warn(
/home/xiaoranzhang/anaconda3/envs/torch_env/lib/python3.9/site-
packages/sklearn/manifold/_t_sne.py:790: FutureWarning: The default learning
rate in TSNE will change from 200.0 to 'auto' in 1.2.
  warnings.warn(
/home/xiaoranzhang/anaconda3/envs/torch_env/lib/python3.9/site-
packages/sklearn/manifold/_t_sne.py:780: FutureWarning: The default
initialization in TSNE will change from 'random' to 'pca' in 1.2.
  warnings.warn(
```

```
/home/xiaoranzhang/anaconda3/envs/torch_env/lib/python3.9/site-
packages/sklearn/manifold/_t_sne.py:790: FutureWarning: The default learning
rate in TSNE will change from 200.0 to 'auto' in 1.2.
  warnings.warn(
/home/xiaoranzhang/anaconda3/envs/torch_env/lib/python3.9/site-
packages/sklearn/manifold/_t_sne.py:780: FutureWarning: The default
initialization in TSNE will change from 'random' to 'pca' in 1.2.
  warnings.warn(
/home/xiaoranzhang/anaconda3/envs/torch_env/lib/python3.9/site-
packages/sklearn/manifold/_t_sne.py:790: FutureWarning: The default learning
rate in TSNE will change from 200.0 to 'auto' in 1.2.
  warnings.warn(
/home/xiaoranzhang/anaconda3/envs/torch_env/lib/python3.9/site-
packages/sklearn/manifold/_t_sne.py:780: FutureWarning: The default
initialization in TSNE will change from 'random' to 'pca' in 1.2.
  warnings.warn(
/home/xiaoranzhang/anaconda3/envs/torch_env/lib/python3.9/site-
packages/sklearn/manifold/_t_sne.py:790: FutureWarning: The default learning
rate in TSNE will change from 200.0 to 'auto' in 1.2.
  warnings.warn(
/home/xiaoranzhang/anaconda3/envs/torch_env/lib/python3.9/site-
packages/sklearn/manifold/_t_sne.py:780: FutureWarning: The default
initialization in TSNE will change from 'random' to 'pca' in 1.2.
  warnings.warn(
/home/xiaoranzhang/anaconda3/envs/torch_env/lib/python3.9/site-
packages/sklearn/manifold/_t_sne.py:790: FutureWarning: The default learning
rate in TSNE will change from 200.0 to 'auto' in 1.2.
  warnings.warn(
/home/xiaoranzhang/anaconda3/envs/torch_env/lib/python3.9/site-
packages/sklearn/manifold/_t_sne.py:780: FutureWarning: The default
initialization in TSNE will change from 'random' to 'pca' in 1.2.
  warnings.warn(
/home/xiaoranzhang/anaconda3/envs/torch_env/lib/python3.9/site-
packages/sklearn/manifold/_t_sne.py:790: FutureWarning: The default learning
rate in TSNE will change from 200.0 to 'auto' in 1.2.
  warnings.warn(
/home/xiaoranzhang/anaconda3/envs/torch_env/lib/python3.9/site-
packages/sklearn/manifold/_t_sne.py:780: FutureWarning: The default
initialization in TSNE will change from 'random' to 'pca' in 1.2.
  warnings.warn(
/home/xiaoranzhang/anaconda3/envs/torch_env/lib/python3.9/site-
packages/sklearn/manifold/_t_sne.py:790: FutureWarning: The default learning
rate in TSNE will change from 200.0 to 'auto' in 1.2.
  warnings.warn(
/home/xiaoranzhang/anaconda3/envs/torch_env/lib/python3.9/site-
packages/sklearn/manifold/_t_sne.py:780: FutureWarning: The default
initialization in TSNE will change from 'random' to 'pca' in 1.2.
  warnings.warn(
```

```
/home/xiaoranzhang/anaconda3/envs/torch_env/lib/python3.9/site-
packages/sklearn/manifold/_t_sne.py:790: FutureWarning: The default learning
rate in TSNE will change from 200.0 to 'auto' in 1.2.
  warnings.warn(
/home/xiaoranzhang/anaconda3/envs/torch_env/lib/python3.9/site-
packages/sklearn/manifold/_t_sne.py:780: FutureWarning: The default
initialization in TSNE will change from 'random' to 'pca' in 1.2.
  warnings.warn(
/home/xiaoranzhang/anaconda3/envs/torch_env/lib/python3.9/site-
packages/sklearn/manifold/_t_sne.py:790: FutureWarning: The default learning
rate in TSNE will change from 200.0 to 'auto' in 1.2.
  warnings.warn(
/home/xiaoranzhang/anaconda3/envs/torch_env/lib/python3.9/site-
packages/sklearn/manifold/_t_sne.py:780: FutureWarning: The default
initialization in TSNE will change from 'random' to 'pca' in 1.2.
  warnings.warn(
/home/xiaoranzhang/anaconda3/envs/torch_env/lib/python3.9/site-
packages/sklearn/manifold/_t_sne.py:790: FutureWarning: The default learning
rate in TSNE will change from 200.0 to 'auto' in 1.2.
  warnings.warn(
/home/xiaoranzhang/anaconda3/envs/torch_env/lib/python3.9/site-
packages/sklearn/manifold/_t_sne.py:780: FutureWarning: The default
initialization in TSNE will change from 'random' to 'pca' in 1.2.
  warnings.warn(
/home/xiaoranzhang/anaconda3/envs/torch_env/lib/python3.9/site-
packages/sklearn/manifold/_t_sne.py:790: FutureWarning: The default learning
rate in TSNE will change from 200.0 to 'auto' in 1.2.
  warnings.warn(
/home/xiaoranzhang/anaconda3/envs/torch_env/lib/python3.9/site-
packages/sklearn/manifold/_t_sne.py:780: FutureWarning: The default
initialization in TSNE will change from 'random' to 'pca' in 1.2.
  warnings.warn(
/home/xiaoranzhang/anaconda3/envs/torch_env/lib/python3.9/site-
packages/sklearn/manifold/_t_sne.py:790: FutureWarning: The default learning
rate in TSNE will change from 200.0 to 'auto' in 1.2.
  warnings.warn(
/home/xiaoranzhang/anaconda3/envs/torch_env/lib/python3.9/site-
packages/sklearn/manifold/_t_sne.py:780: FutureWarning: The default
initialization in TSNE will change from 'random' to 'pca' in 1.2.
  warnings.warn(
/home/xiaoranzhang/anaconda3/envs/torch_env/lib/python3.9/site-
packages/sklearn/manifold/_t_sne.py:790: FutureWarning: The default learning
rate in TSNE will change from 200.0 to 'auto' in 1.2.
  warnings.warn(
/home/xiaoranzhang/anaconda3/envs/torch_env/lib/python3.9/site-
packages/sklearn/manifold/_t_sne.py:780: FutureWarning: The default
initialization in TSNE will change from 'random' to 'pca' in 1.2.
  warnings.warn(
```

```
/home/xiaoranzhang/anaconda3/envs/torch_env/lib/python3.9/site-
packages/sklearn/manifold/_t_sne.py:790: FutureWarning: The default learning
rate in TSNE will change from 200.0 to 'auto' in 1.2.
  warnings.warn(
/home/xiaoranzhang/anaconda3/envs/torch_env/lib/python3.9/site-
packages/sklearn/manifold/_t_sne.py:780: FutureWarning: The default
initialization in TSNE will change from 'random' to 'pca' in 1.2.
  warnings.warn(
/home/xiaoranzhang/anaconda3/envs/torch_env/lib/python3.9/site-
packages/sklearn/manifold/_t_sne.py:790: FutureWarning: The default learning
rate in TSNE will change from 200.0 to 'auto' in 1.2.
  warnings.warn(
/home/xiaoranzhang/anaconda3/envs/torch_env/lib/python3.9/site-
packages/sklearn/manifold/_t_sne.py:780: FutureWarning: The default
initialization in TSNE will change from 'random' to 'pca' in 1.2.
  warnings.warn(
/home/xiaoranzhang/anaconda3/envs/torch_env/lib/python3.9/site-
packages/sklearn/manifold/_t_sne.py:790: FutureWarning: The default learning
rate in TSNE will change from 200.0 to 'auto' in 1.2.
  warnings.warn(
/home/xiaoranzhang/anaconda3/envs/torch_env/lib/python3.9/site-
packages/sklearn/manifold/_t_sne.py:780: FutureWarning: The default
initialization in TSNE will change from 'random' to 'pca' in 1.2.
  warnings.warn(
/home/xiaoranzhang/anaconda3/envs/torch_env/lib/python3.9/site-
packages/sklearn/manifold/_t_sne.py:790: FutureWarning: The default learning
rate in TSNE will change from 200.0 to 'auto' in 1.2.
  warnings.warn(
```


Trial 1
Trial 100
Trial 200
Trial 300
Trial 400
Wavelet db1+SVM
Wavelet db2+SVM
Wavelet db3+SVM
Wavelet db4+SVM
t-SNE+SVM
Shallow CNN