tSNE

In [1]:

```
import sys
sys.path.insert(1, '../../_tools/')
import torch as th
import numpy as np
import torchvision
import torchvision.transforms as trasf
import matplotlib.pyplot as plt

seed = 42
th.manual_seed(seed)
np.random.seed(seed)
device = th.device('cpu')
```

Data

In [2]:

```
TS_torch_dataset = torchvision.datasets.MNIST(
    root=r"C:\Users\matte\LocalData\Master Thesis",
    train=False,
    download=True,
    transform=trasf.ToTensor()
)

# Load everything
TSx = th.stack([x.flatten() for x, y in TS_torch_dataset]).to(device)
TSy = th.tensor([y for x, y in TS_torch_dataset], device=device)
TSy = TSy.numpy()
```

Models

In [3]:

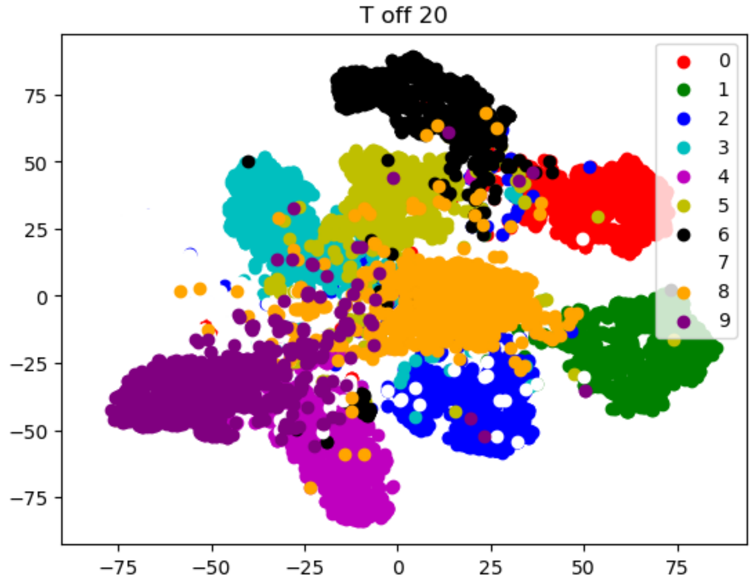
```
base_path = "../../mnist experiment/final models/"

models = {
    'T off 20': None,
    'T on 20': None,
    'T off 200': None,
    'T on 200': None,
    'T off 2000': None,
    'T on 2000': None
}

for exp in models:
    models[exp] = th.load(base_path+exp+'.pt', device)
```

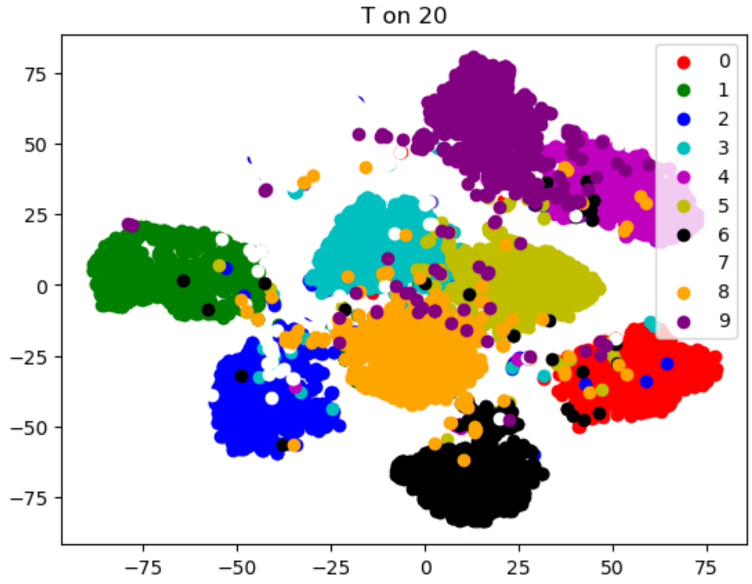
In [4]:

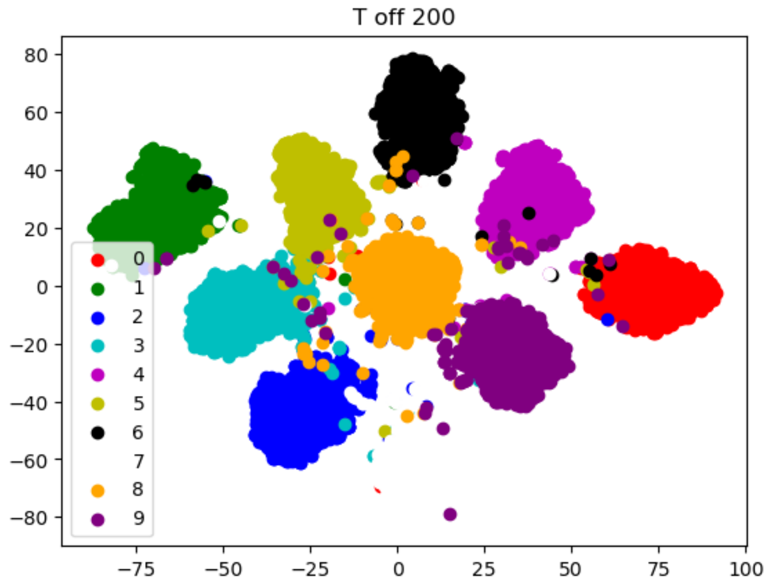
```
from sklearn.manifold import TSNE
import pickle

exps_x2D = {}

for exp, model in models.items():
    print(exp)
    tsne = TSNE(learning_rate='auto', init='pca', n_iter=1000, verbose=1)
    sum_norm_godness = model.sum_normalized_goodness(TSx)
    exps_x2D[exp] = tsne.fit_transform(sum_norm_godness)

with open("exps_x2D.pickle", 'wb') as f:
    pickle.dump(exps_x2D, f)
```

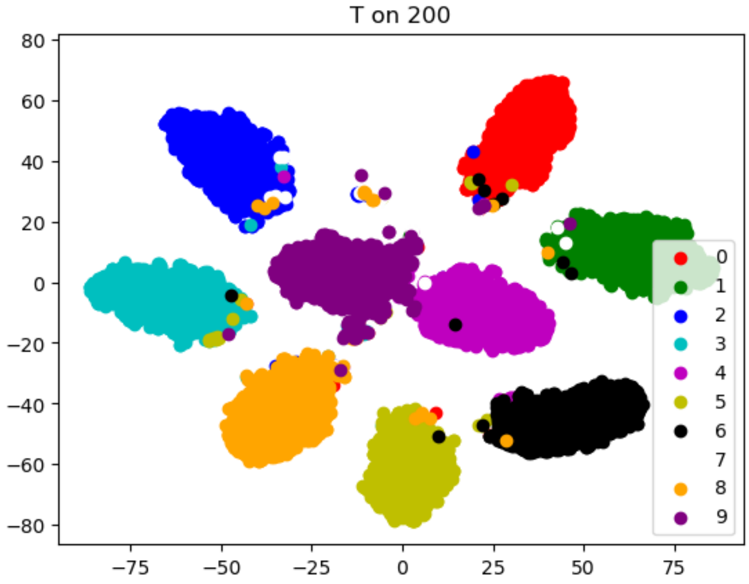
In [5]:

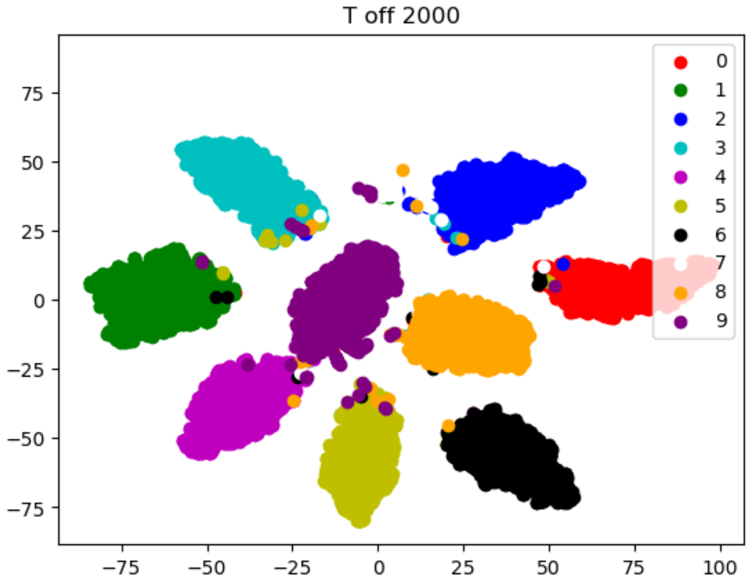
```
with open("exps_x2D.pickle", 'rb') as f:
    exps_x2D = pickle.load(f)

labels = list(range(0, 10))
colors = ['r', 'g', 'b', 'c', 'm', 'y', 'k', 'w', 'orange', 'purple']
names = [str(label) for label in labels]

for exp, x2D in exps_x2D.items():

    plt.figure()
    plt.title(exp)
    for label, color, name in zip(labels, colors, names):
        x = x2D[TSy==label]
        plt.scatter(x[:, 0], x[:, 1], color=color, label=name)
    plt.legend(loc='best')
```

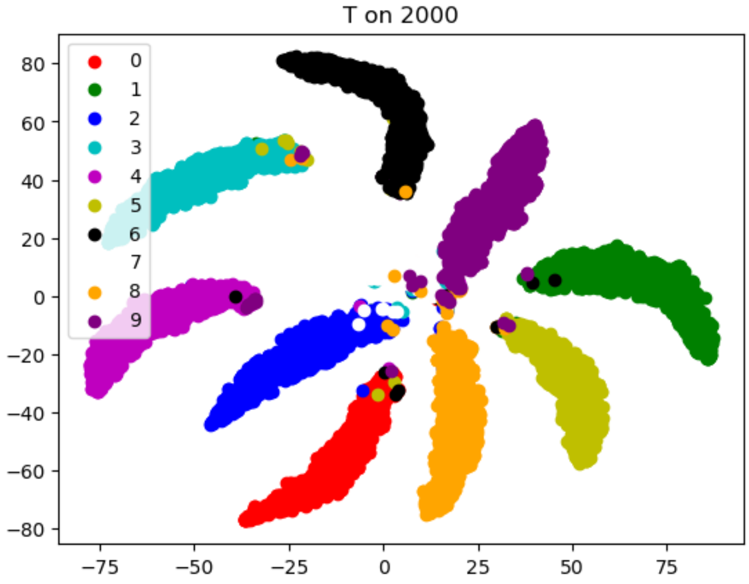

T off 20
75
50
25
0
−25
−50
−75
−75
−50
−25
0
25
50
75
● 0
● 1
● 2
● 3
● 4
● 5
● 6
7
● 8
● 9


T on 20
75
50
25
0
−25
−50
−75
−75
−50
−25
0
25
50
75
● 0
● 1
● 2
● 3
● 4
● 5
● 6
7
● 8
● 9


T off 200
80
60
40
20
0
−20
−40
−60
−80
−75
−50
−25
0
25
50
75
100
● 0
● 1
● 2
● 3
● 4
● 5
● 6
7
● 8
● 9


T on 200


T off 2000


T on 2000