In [7]:
```
!pip install torch torchvision matplotlib
```

```
Requirement already satisfied: torch in /usr/local/lib/python3.12/dist-packages
(2.9.0+cu126)
Requirement already satisfied: torchvision in /usr/local/lib/python3.12/dist-pa
ckages (0.24.0+cu126)
Requirement already satisfied: matplotlib in /usr/local/lib/python3.12/dist-pac
kages (3.10.0)
Requirement already satisfied: filelock in /usr/local/lib/python3.12/dist-packa
ges (from torch) (3.20.3)
Requirement already satisfied: typing-extensions>=4.10.0 in /usr/local/lib/pyth
on3.12/dist-packages (from torch) (4.15.0)
Requirement already satisfied: setuptools in /usr/local/lib/python3.12/dist-pac
kages (from torch) (75.2.0)
Requirement already satisfied: sympy>=1.13.3 in /usr/local/lib/python3.12/dist-
packages (from torch) (1.14.0)
Requirement already satisfied: networkx>=2.5.1 in /usr/local/lib/python3.12/dis
t-packages (from torch) (3.6.1)
Requirement already satisfied: jinja2 in /usr/local/lib/python3.12/dist-package
s (from torch) (3.1.6)
Requirement already satisfied: fsspec>=0.8.5 in /usr/local/lib/python3.12/dist-
packages (from torch) (2025.3.0)
Requirement already satisfied: nvidia-cuda-nvrtc-cu12==12.6.77 in /usr/local/li
b/python3.12/dist-packages (from torch) (12.6.77)
Requirement already satisfied: nvidia-cuda-runtime-cu12==12.6.77 in /usr/local/
lib/python3.12/dist-packages (from torch) (12.6.77)
Requirement already satisfied: nvidia-cuda-cupti-cu12==12.6.80 in /usr/local/li
b/python3.12/dist-packages (from torch) (12.6.80)
Requirement already satisfied: nvidia-cudnn-cu12==9.10.2.21 in /usr/local/lib/p
ython3.12/dist-packages (from torch) (9.10.2.21)
Requirement already satisfied: nvidia-cublas-cu12==12.6.4.1 in /usr/local/lib/p
ython3.12/dist-packages (from torch) (12.6.4.1)
Requirement already satisfied: nvidia-cufft-cu12==11.3.0.4 in /usr/local/lib/py
thon3.12/dist-packages (from torch) (11.3.0.4)
Requirement already satisfied: nvidia-curand-cu12==10.3.7.77 in /usr/local/lib/
python3.12/dist-packages (from torch) (10.3.7.77)
Requirement already satisfied: nvidia-cusolver-cu12==11.7.1.2 in /usr/local/li
b/python3.12/dist-packages (from torch) (11.7.1.2)
Requirement already satisfied: nvidia-cusparse-cu12==12.5.4.2 in /usr/local/li
b/python3.12/dist-packages (from torch) (12.5.4.2)
Requirement already satisfied: nvidia-cusparselt-cu12==0.7.1 in /usr/local/lib/
python3.12/dist-packages (from torch) (0.7.1)
Requirement already satisfied: nvidia-nccl-cu12==2.27.5 in /usr/local/lib/pytho
n3.12/dist-packages (from torch) (2.27.5)
Requirement already satisfied: nvidia-nvshmem-cu12==3.3.20 in /usr/local/lib/py
thon3.12/dist-packages (from torch) (3.3.20)
Requirement already satisfied: nvidia-nvtx-cu12==12.6.77 in /usr/local/lib/pyth
on3.12/dist-packages (from torch) (12.6.77)
Requirement already satisfied: nvidia-nvjitlink-cu12==12.6.85 in /usr/local/li
b/python3.12/dist-packages (from torch) (12.6.85)
Requirement already satisfied: nvidia-cufile-cu12==1.11.1.6 in /usr/local/lib/p
ython3.12/dist-packages (from torch) (1.11.1.6)
Requirement already satisfied: triton==3.5.0 in /usr/local/lib/python3.12/dist-
packages (from torch) (3.5.0)
Requirement already satisfied: numpy in /usr/local/lib/python3.12/dist-packages
(from torchvision) (2.0.2)
```

```
Requirement already satisfied: pillow!=8.3.*,>=5.3.0 in /usr/local/lib/python
3.12/dist-packages (from torchvision) (11.3.0)
Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/python3.12/di
st-packages (from matplotlib) (1.3.3)
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.12/dist-p
ackages (from matplotlib) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.12/d
ist-packages (from matplotlib) (4.61.1)
Requirement already satisfied: kiwisolver>=1.3.1 in /usr/local/lib/python3.12/d
ist-packages (from matplotlib) (1.4.9)
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.12/dis
t-packages (from matplotlib) (25.0)
Requirement already satisfied: pyparsing>=2.3.1 in /usr/local/lib/python3.12/di
st-packages (from matplotlib) (3.3.2)
Requirement already satisfied: python-dateutil>=2.7 in /usr/local/lib/python3.1
2/dist-packages (from matplotlib) (2.9.0.post0)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.12/dist-packa
ges (from python-dateutil>=2.7->matplotlib) (1.17.0)
Requirement already satisfied: mpmath<1.4,>=1.1.0 in /usr/local/lib/python3.12/
dist-packages (from sympy>=1.13.3->torch) (1.3.0)
Requirement already satisfied: MarkupSafe>=2.0 in /usr/local/lib/python3.12/dis
t-packages (from jinja2->torch) (3.0.3)
```

In [10]:
```python
import torch
import torch.nn as nn
import torch.optim as optim
import torchvision
import matplotlib
from torchvision import datasets, transforms
from torch.utils.data import DataLoader
import matplotlib.pyplot as plt

print("Torch version:", torch.__version__)
print("Torchvision version:", torchvision.__version__)
print("Matplotlib version:", matplotlib.__version__)
print("CUDA available:", torch.cuda.is_available())
```

```
Torch version: 2.9.0+cu126
Torchvision version: 0.24.0+cu126
Matplotlib version: 3.10.0
CUDA available: True
```

In [11]:
```python
transform = transforms.Compose([
    transforms.ToTensor()
])

train_dataset = datasets.MNIST(root='./data', train=True, download=True, trans
test_dataset = datasets.MNIST(root='./data', train=False, download=True, trans

train_loader = DataLoader(train_dataset, batch_size=128, shuffle=True)
test_loader = DataLoader(test_dataset, batch_size=128, shuffle=False)
```

```
100%|██████████| 9.91M/9.91M [00:00<00:00, 12.6MB/s]
100%|██████████| 28.9k/28.9k [00:00<00:00, 337kB/s]
100%|██████████| 1.65M/1.65M [00:00<00:00, 3.18MB/s]
100%|██████████| 4.54k/4.54k [00:00<00:00, 15.4MB/s]
```

In [12]:
```
class VAE(nn.Module):
    def __init__(self, latent_dim=20):
        super().__init__()

        self.encoder = nn.Sequential(
            nn.Linear(28*28, 400),
            nn.ReLU()
        )

        self.mu = nn.Linear(400, latent_dim)
        self.logvar = nn.Linear(400, latent_dim)

        self.decoder = nn.Sequential(
            nn.Linear(latent_dim, 400),
            nn.ReLU(),
            nn.Linear(400, 28*28),
            nn.Sigmoid()
        )

    def reparameterize(self, mu, logvar):
        std = torch.exp(0.5 * logvar)
        eps = torch.randn_like(std)
        return mu + eps * std

    def forward(self, x):
        x = x.view(-1, 28*28)
        h = self.encoder(x)
        mu = self.mu(h)
        logvar = self.logvar(h)
        z = self.reparameterize(mu, logvar)
        recon = self.decoder(z)
        return recon, mu, logvar
```

In [13]:
```
def vae_loss(recon_x, x, mu, logvar):
    recon_loss = nn.functional.binary_cross_entropy(
        recon_x, x.view(-1, 28*28), reduction='sum'
    )

    kl_loss = -0.5 * torch.sum(1 + logvar - mu.pow(2) - logvar.exp())

    return recon_loss + kl_loss
```

In [14]:
```
device = torch.device("cuda" if torch.cuda.is_available() else "cpu")

model = VAE(latent_dim=20).to(device)
optimizer = optim.Adam(model.parameters(), lr=1e-3)

epochs = 200
```

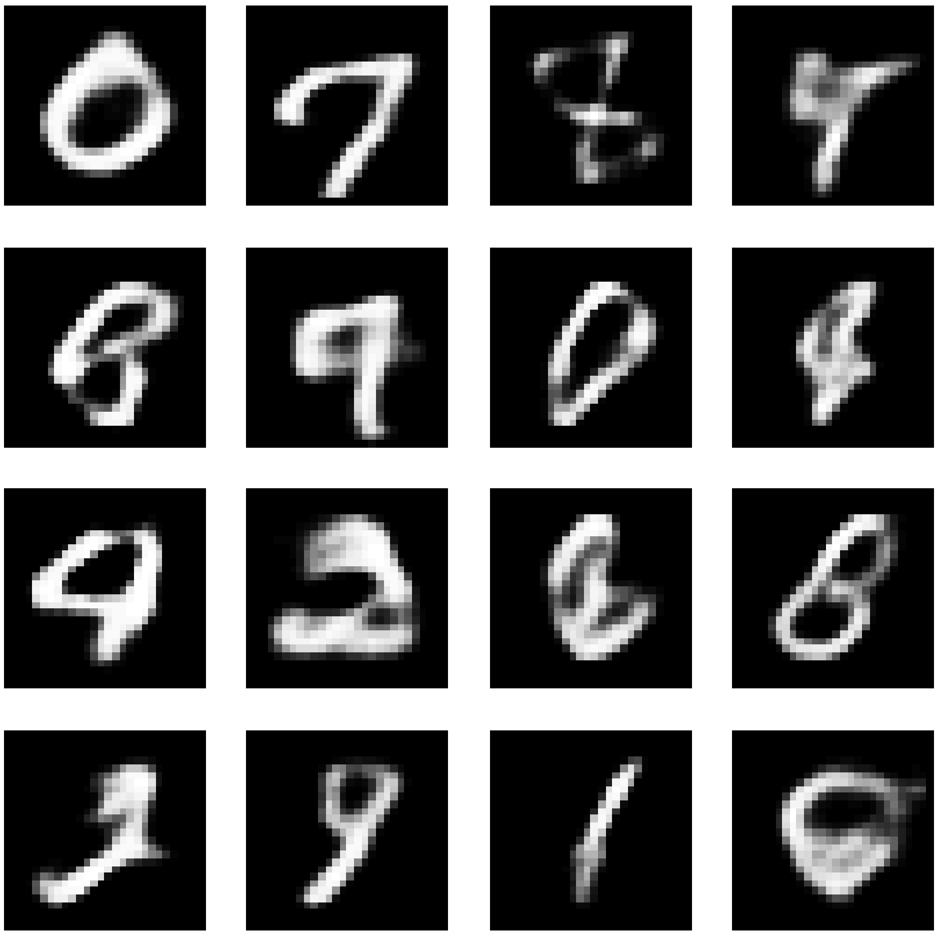
```python
train_losses = []

for epoch in range(epochs):
    model.train()
    total_loss = 0

    for x, _ in train_loader:
        x = x.to(device)
        optimizer.zero_grad()

        recon, mu, logvar = model(x)
        loss = vae_loss(recon, x, mu, logvar)

        loss.backward()
        optimizer.step()

        total_loss += loss.item()

    avg_loss = total_loss / len(train_loader.dataset)
    train_losses.append(avg_loss)
    print(f"Epoch {epoch+1}/{epochs}, Loss: {avg_loss:.4f}")
```

```
Epoch 1/200, Loss: 164.2397
Epoch 2/200, Loss: 121.1511
Epoch 3/200, Loss: 114.2985
Epoch 4/200, Loss: 111.3156
Epoch 5/200, Loss: 109.5882
Epoch 6/200, Loss: 108.3811
Epoch 7/200, Loss: 107.5304
Epoch 8/200, Loss: 106.9180
Epoch 9/200, Loss: 106.3820
Epoch 10/200, Loss: 105.9730
Epoch 11/200, Loss: 105.6190
Epoch 12/200, Loss: 105.3317
Epoch 13/200, Loss: 105.0971
Epoch 14/200, Loss: 104.7848
Epoch 15/200, Loss: 104.6277
Epoch 16/200, Loss: 104.4344
Epoch 17/200, Loss: 104.2428
Epoch 18/200, Loss: 104.0961
Epoch 19/200, Loss: 103.9321
Epoch 20/200, Loss: 103.8248
Epoch 21/200, Loss: 103.6477
Epoch 22/200, Loss: 103.5562
Epoch 23/200, Loss: 103.4211
Epoch 24/200, Loss: 103.3496
Epoch 25/200, Loss: 103.2152
Epoch 26/200, Loss: 103.1174
Epoch 27/200, Loss: 102.9989
Epoch 28/200, Loss: 102.9521
Epoch 29/200, Loss: 102.9031
Epoch 30/200, Loss: 102.7609
Epoch 31/200, Loss: 102.7209
Epoch 32/200, Loss: 102.6225
Epoch 33/200, Loss: 102.5963
Epoch 34/200, Loss: 102.5299
Epoch 35/200, Loss: 102.4413
Epoch 36/200, Loss: 102.3987
Epoch 37/200, Loss: 102.3204
Epoch 38/200, Loss: 102.2436
Epoch 39/200, Loss: 102.1793
Epoch 40/200, Loss: 102.1760
Epoch 41/200, Loss: 102.1147
Epoch 42/200, Loss: 102.0636
Epoch 43/200, Loss: 101.9767
Epoch 44/200, Loss: 101.9641
Epoch 45/200, Loss: 101.9116
Epoch 46/200, Loss: 101.8591
Epoch 47/200, Loss: 101.8514
Epoch 48/200, Loss: 101.7711
Epoch 49/200, Loss: 101.7278
Epoch 50/200, Loss: 101.6726
Epoch 51/200, Loss: 101.6407
Epoch 52/200, Loss: 101.6122
Epoch 53/200, Loss: 101.6244
Epoch 54/200, Loss: 101.5572
Epoch 55/200, Loss: 101.5303
Epoch 56/200, Loss: 101.4419
Epoch 57/200, Loss: 101.4287
Epoch 58/200, Loss: 101.3748
Epoch 59/200, Loss: 101.3958
Epoch 60/200, Loss: 101.3609
Epoch 61/200, Loss: 101.3131
Epoch 62/200, Loss: 101.2478
Epoch 63/200, Loss: 101.2603
Epoch 64/200, Loss: 101.2411
Epoch 65/200, Loss: 101.1898
Epoch 66/200, Loss: 101.1522
Epoch 67/200, Loss: 101.1470
Epoch 68/200, Loss: 101.1268
Epoch 69/200, Loss: 101.0564
Epoch 70/200, Loss: 101.0891
Epoch 71/200, Loss: 101.0163
Epoch 72/200, Loss: 100.9925
Epoch 73/200, Loss: 100.9849
Epoch 74/200, Loss: 100.9381
Epoch 75/200, Loss: 100.9364
Epoch 76/200, Loss: 100.9108
Epoch 77/200, Loss: 100.8821
Epoch 78/200, Loss: 100.8782
Epoch 79/200, Loss: 100.8450
Epoch 80/200, Loss: 100.8164
Epoch 81/200, Loss: 100.8113
Epoch 82/200, Loss: 100.7775
Epoch 83/200, Loss: 100.7306
Epoch 84/200, Loss: 100.7526
Epoch 85/200, Loss: 100.6467
Epoch 86/200, Loss: 100.7195
Epoch 87/200, Loss: 100.6598
Epoch 88/200, Loss: 100.6752
Epoch 89/200, Loss: 100.6986
Epoch 90/200, Loss: 100.6208
Epoch 91/200, Loss: 100.6116
Epoch 92/200, Loss: 100.5970
Epoch 93/200, Loss: 100.5787
Epoch 94/200, Loss: 100.5608
Epoch 95/200, Loss: 100.5315
Epoch 96/200, Loss: 100.5144
Epoch 97/200, Loss: 100.4889
Epoch 98/200, Loss: 100.4694
Epoch 99/200, Loss: 100.4438
Epoch 100/200, Loss: 100.4310
Epoch 101/200, Loss: 100.4213
Epoch 102/200, Loss: 100.3953
Epoch 103/200, Loss: 100.4000
Epoch 104/200, Loss: 100.3763
Epoch 105/200, Loss: 100.3363
Epoch 106/200, Loss: 100.3287
Epoch 107/200, Loss: 100.3350
```

```
---------------------------------------------------------------------------
KeyboardInterrupt                         Traceback (most recent call last)
/tmp/ipython-input-2909306000.py in <cell line: 0>()
     11     total_loss = 0
     12 
---> 13     for x, _ in train_loader:
     14         x = x.to(device)
     15         optimizer.zero_grad()

/usr/local/lib/python3.12/dist-packages/torch/utils/data/dataloader.py in __nex
t__(self)
    730                 # TODO(https://github.com/pytorch/pytorch/issues/76750)
    731                 self._reset()  # type: ignore[call-arg]
--> 732             data = self._next_data()
    733             self._num_yielded += 1
    734             if (

/usr/local/lib/python3.12/dist-packages/torch/utils/data/dataloader.py in _nex
t_data(self)
    786     def _next_data(self):
    787         index = self._next_index()  # may raise StopIteration
--> 788         data = self._dataset_fetcher.fetch(index)  # may raise StopIter
ation
    789         if self._pin_memory:
    790             data = _utils.pin_memory.pin_memory(data, self._pin_memor
y_device)

/usr/local/lib/python3.12/dist-packages/torch/utils/data/_utils/fetch.py in fet
ch(self, possibly_batched_index)
     50                 data = self.dataset.__getitems__(possibly_batched_inde
x)
     51             else:
---> 52                 data = [self.dataset[idx] for idx in possibly_batched_i
ndex]
     53         else:
     54             data = self.dataset[possibly_batched_index]

/usr/local/lib/python3.12/dist-packages/torchvision/datasets/mnist.py in __geti
tem__(self, index)
    144 
    145         if self.transform is not None:
--> 146             img = self.transform(img)
    147 
    148         if self.target_transform is not None:

/usr/local/lib/python3.12/dist-packages/torchvision/transforms/transforms.py in
__call__(self, img)
     93     def __call__(self, img):
     94         for t in self.transforms:
---> 95             img = t(img)
     96         return img
     97 

/usr/local/lib/python3.12/dist-packages/torchvision/transforms/transforms.py in
```

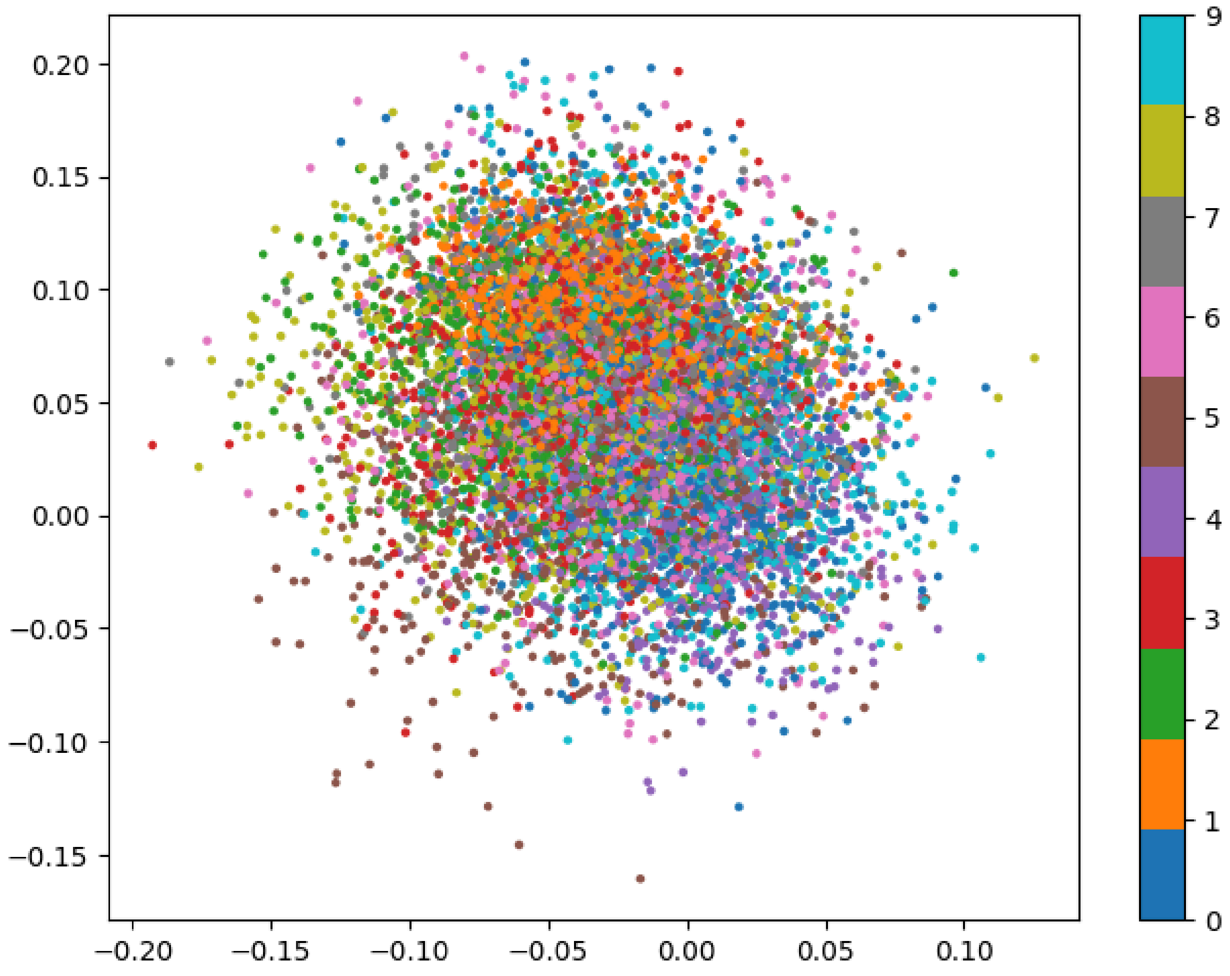
```
__call__(self, pic)
    135             Tensor: Converted image.
    136         """
--> 137         return F.to_tensor(pic)
    138 
    139     def __repr__(self) -> str:

/usr/local/lib/python3.12/dist-packages/torchvision/transforms/functional.py in
to_tensor(pic)
    166     # handle PIL Image
    167     mode_to_nptype = {"I": np.int32, "I;16" if sys.byteorder == "littl
e" else "I;16B": np.int16, "F": np.float32}
--> 168     img = torch.from_numpy(np.array(pic, mode_to_nptype.get(pic.mode, n
p.uint8), copy=True))
    169 
    170     if pic.mode == "1":

/usr/local/lib/python3.12/dist-packages/PIL/Image.py in __array_interface__(sel
f)
    733             new["data"] = self.tobytes("raw", "L")
    734         else:
--> 735             new["data"] = self.tobytes()
    736         new["shape"], new["typestr"] = _conv_type_shape(self)
    737         return new

/usr/local/lib/python3.12/dist-packages/PIL/Image.py in tobytes(self, encoder_n
ame, *args)
    800 
    801         # unpack data
--> 802         e = _getencoder(self.mode, encoder_name, encoder_args)
    803         e.setimage(self.im)
    804 

/usr/local/lib/python3.12/dist-packages/PIL/Image.py in _getencoder(mode, encod
er_name, args, extra)
    459         msg = f"encoder {encoder_name} not available"
    460         raise OSError(msg) from e
--> 461     return encoder(mode, *args + extra)
    462 
    463 

KeyboardInterrupt:
```

In [15]:
```python
model.eval()
with torch.no_grad():
    z = torch.randn(16, 20).to(device)
    samples = model.decoder(z).view(-1, 1, 28, 28).cpu()

plt.figure(figsize=(6,6))
for i in range(16):
    plt.subplot(4,4,i+1)
    plt.imshow(samples[i][0], cmap='gray')
    plt.axis('off')
plt.show()
```



In [16]:
```
model_2d = VAE(latent_dim=2).to(device)
optimizer = optim.Adam(model_2d.parameters(), lr=1e-3)
```

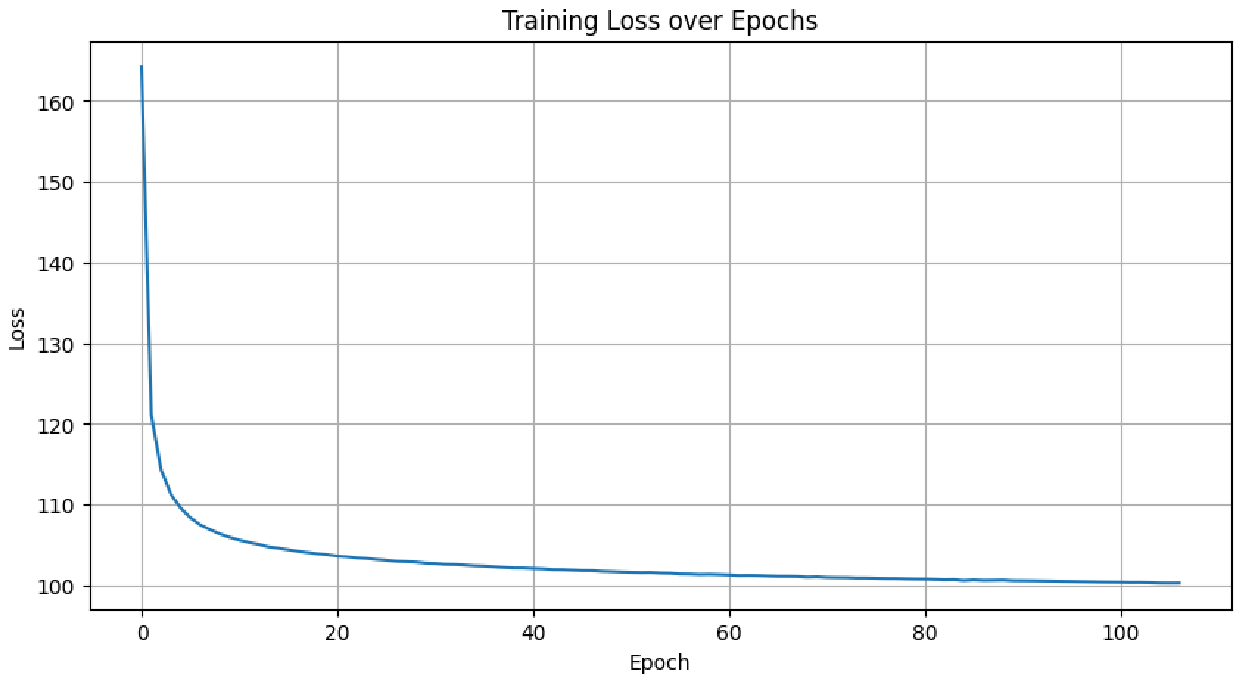
In [17]:
```
model_2d.eval()
latents = []
labels = []

with torch.no_grad():
    for x, y in test_loader:
        x = x.to(device)
        _, mu, _ = model_2d(x)
        latents.append(mu.cpu())
        labels.append(y)

latents = torch.cat(latents)
labels = torch.cat(labels)

plt.figure(figsize=(8,6))
plt.scatter(latents[:,0], latents[:,1], c=labels, cmap='tab10', s=5)
plt.colorbar()
plt.show()
```



```
In [18]: import os

         output_dir = './saved_models'
         os.makedirs(output_dir, exist_ok=True)

         model_path = os.path.join(output_dir, 'vae_model.pth')
         torch.save(model.state_dict(), model_path)

         print(f"Model saved to: {model_path}")
```

```
Model saved to: ./saved_models/vae_model.pth
```

```
In [19]: plt.figure(figsize=(10, 5))
         plt.plot(train_losses)
         plt.title('Training Loss over Epochs')
         plt.xlabel('Epoch')
         plt.ylabel('Loss')
         plt.grid(True)
         plt.show()
```


Training Loss over Epochs
Loss
Epoch
160
150
140
130
120
110
100
0
20
40
60
80
100

```
In [ ]:
```