Project 5

by Jiangyong Huang on Dec 20, 2022

Integrating Generative and Descriptive Model

Generative model augmented with latent space modeled by Descriptive EBM

[ ]:
```
import os
import torch
import numpy as np
import torch.nn as nn
from PIL import Image
from matplotlib import pyplot as plt
from torch.nn import functional as F
from torchvision import transforms as T
from torchvision.utils import make_grid
%matplotlib inline
torch.manual_seed(1220)
torch.cuda.manual_seed(1220)
```

Configs

[ ]:
```
class Config(object):
    def __init__(self):
        self.device = 'cuda' if torch.cuda.is_available() else 'cpu'
        self.langevin_steps = 20
        self.langevin_step_size = 0.1
        self.sigma = 1
        self.prior_sigma = 1
        self.epochs = 3000
        self.lr = 1e-3
        self.interpolation_d = 5
        self.z_dim = 32

args = Config()
```

Dataset

[ ]:
```
def load_data(data_dir='images'):
    imgs = []
    transform = T.Compose([T.Resize([128, 128]), T.ToTensor(), T.Normalize((0.
 ↪5, 0.5, 0.5), (0.5, 0.5, 0.5))])

    for fname in os.listdir(data_dir):
        img = Image.open(os.path.join(data_dir, fname)).convert('RGB')
        img = transform(img)
        imgs.append(img)

    imgs = torch.stack(imgs)
    return imgs
```

Model

[ ]:
```
def set_cuda(deterministic=True):
    if torch.cuda.is_available():
        if deterministic:
            torch.backends.cudnn.deterministic = True
            torch.backends.cudnn.benchmark = False
        else:
            torch.backends.cudnn.deterministic = False
            torch.backends.cudnn.benchmark = True


def make_upsample_block(in_dim, out_dim, is_last=False):
    if is_last:
        return nn.Sequential(
            nn.ConvTranspose2d(in_channels=in_dim, out_channels=out_dim,␣
 ↪kernel_size=4, stride=2, padding=1),
            nn.BatchNorm2d(num_features=out_dim),
            nn.ReLU(inplace=True),
            nn.Conv2d(in_channels=out_dim, out_channels=3, kernel_size=3,␣
 ↪stride=1, padding=1),
            nn.Tanh()
        )
    else:
        return nn.Sequential(
            nn.ConvTranspose2d(in_channels=in_dim, out_channels=out_dim,␣
 ↪kernel_size=4, stride=2, padding=1),
            nn.BatchNorm2d(num_features=out_dim),
            nn.ReLU(inplace=True)
        )


class DesNet(nn.Module):
    def __init__(self, z_dim, hidden_dim):
```

```
        """
        Instead of encoding image to latent like VAE,
        this descriptor directly models the prior distribution of latent␣
↪variable z.
        """
        super().__init__()
        self.layer1 = nn.Linear(z_dim, hidden_dim)
        self.layer2 = nn.Linear(hidden_dim, hidden_dim)
        self.layer3 = nn.Linear(hidden_dim, 1)

    def forward(self, z):
        z = z.to(self.layer1.weight.device)
        x = self.layer1(z)
        x = F.relu(x)
        x = self.layer2(x)
        x = F.relu(x)
        x = self.layer3(x)
        return x.squeeze(-1)


class GenNet(nn.Module):
    def __init__(self, z_dim):
        """
        Decode latent variable z to form an output image at 128x128.
        The latent z is first projected to feature map at 2x2, which further␣
↪goes through transposed conv to 128x128.
        (z_dim) -> (512, 4, 4) -> (512, 8, 8) -> (256, 16, 16) -> (128, 32, 32)␣
↪-> (64, 64, 64) -> (3, 128, 128)
        """
        super().__init__()
        self.z_proj = nn.Linear(z_dim, 512*4*4)
        self.block1 = make_upsample_block(in_dim=512, out_dim=512,␣
↪is_last=False)
        self.block2 = make_upsample_block(in_dim=512, out_dim=256,␣
↪is_last=False)
        self.block3 = make_upsample_block(in_dim=256, out_dim=128,␣
↪is_last=False)
        self.block4 = make_upsample_block(in_dim=128, out_dim=64, is_last=False)
        self.block5 = make_upsample_block(in_dim=64, out_dim=64, is_last=True)

    def forward(self, z):
        # [-1, z_dim]
        z = z.to(self.z_proj.weight.device)
        x = self.z_proj(z)
        # [-1, 8192]
        x = x.view(-1, 512, 4, 4)
```

```
        x = self.block1(x)
        # [-1, 512, 8, 8]
        x = self.block2(x)
        # [-1, 256, 16, 16]
        x = self.block3(x)
        # [-1, 128, 32, 32]
        x = self.block4(x)
        # [-1, 64, 64, 64]
        x = self.block5(x)
        # [-1, 3, 128, 128]
        return x


def weights_init_xavier(m):
    classname = m.__class__.__name__
    if 'Conv' in classname:
        nn.init.xavier_normal_(m.weight)
    elif 'BatchNorm' in classname:
        m.weight.data.normal_(1, 0.02)
        m.bias.data.fill_(0)


class Model(nn.Module):
    def __init__(self, args):
        super().__init__()
        self.langevin_steps = args.langevin_steps
        self.langevin_step_size = args.langevin_step_size
        self.sigma = args.sigma
        self.prior_sigma = args.prior_sigma
        self.lr = args.lr

        self.Generator = GenNet(z_dim=args.z_dim)
        self.Generator.apply(weights_init_xavier)
        self.Generator.to(args.device)

        self.Descriptor = DesNet(z_dim=args.z_dim, hidden_dim=2*args.z_dim)
        self.Descriptor.apply(weights_init_xavier)
        self.Descriptor.to(args.device)

    def sample_langevin_prior(self, z):
        # prior p(z)
        for i in range(self.langevin_steps):
            z = z.detach()
            z.requires_grad = True

            prior_log_likelihood = self.Descriptor(z) - (z**2).sum() / (2 *␣
 ↪self.prior_sigma**2)
```

```
            z_grad_descriptor = torch.autograd.grad(prior_log_likelihood, z)[0]

            grad = z_grad_descriptor.clone()
            norm = torch.linalg.norm(grad)
            if norm > 1:
                grad /= norm
            noise = torch.randn_like(z, device=z.device)
            z = z + self.langevin_step_size**2 * grad / 2 + self.langevin_step_size * noise

        return z.detach()

    def sample_langevin_post(self, z, I):
        # posterior p(z|I) \propto p(z)p(I|z)
        for i in range(self.langevin_steps):
            z = z.detach()
            z.requires_grad = True

            prior_log_likelihood = self.Descriptor(z) - (z**2).sum() / (2 * self.prior_sigma**2)
            z_grad_descriptor = torch.autograd.grad(prior_log_likelihood, z)[0]

            g = self.Generator(z)
            cond_log_likelihood = -F.mse_loss(g.squeeze(), I.to(g.device), reduction='sum') / (2 * self.sigma**2)
            z_grad_generator = torch.autograd.grad(cond_log_likelihood, z)[0]

            grad = z_grad_descriptor + z_grad_generator
            norm = torch.linalg.norm(grad)
            if norm > 1:
                grad /= norm
            noise = torch.randn_like(z, device=z.device)

            z = z + self.langevin_step_size**2 * grad / 2 + self.langevin_step_size * noise

        return z.detach()

    def update(self, zs_prior, zs_post, imgs):
        """
        latents: [N, z_dim]
        imgs: [N, 3, 128, 128]
        """
        named_grads = {}
        zs_prior_new = []
        zs_post_new = []
        energy_pos = []
```

```
        energy_neg = []
        loss_generator = []
        for z_prior, z_post, I in zip(zs_prior, zs_post, imgs):
            z_prior = self.sample_langevin_prior(z_prior)
            z_post = self.sample_langevin_post(z_post, I)

            self.Descriptor.zero_grad()
            score_pos = self.Descriptor(z_post)
            score_neg = self.Descriptor(z_prior)
            score_descriptor = score_pos - score_neg
            score_descriptor.backward()

            self.Generator.zero_grad()
            g = self.Generator(z_post)
            score_generator = -F.mse_loss(g.squeeze(), I.to(g.device),␣
↪reduction='sum') / (2 * self.sigma**2)
            score_generator.backward()

            # collect gradients
            for n, p in self.named_parameters():
                if n not in named_grads.keys():
                    named_grads.update({n: []})
                named_grads[n].append(p.grad.clone())

            zs_prior_new.append(z_prior.view(-1, z_prior.numel()))
            zs_post_new.append(z_post.view(-1, z_post.numel()))
            # record loss
            energy_pos.append(-score_pos.detach().item())
            energy_neg.append(-score_neg.detach().item())
            loss_generator.append(-score_generator.detach().item())

        # update parameters
        for n, p in self.named_parameters():
            grad = torch.stack(named_grads[n], dim=0).mean(0)
            norm = torch.linalg.norm(grad)
            if norm > 1:
                grad /= norm
            p.data += self.lr * grad

        return torch.cat(zs_prior_new, dim=0).detach(), torch.cat(zs_post_new,␣
↪dim=0).detach(), \
               np.mean(energy_pos), np.mean(energy_neg), np.mean(loss_generator)
```

Visualization tools

[ ]:
```
@torch.no_grad()
def sample_img(model, z):
```

```
    sample = model.Generator(z)
    sample = sample.squeeze().detach().cpu().numpy()
    # [-1, 1] to [0, 1]
    sample = (sample + 1) / 2
    # [C, H, W] to [H, W, C]
    sample = sample.transpose(1, 2, 0)
    return sample


def visualize_img_tensor(img_tensor, title_str):
    if img_tensor.ndim == 4:
        img_tensor = img_tensor[0]
    img_array = img_tensor.detach().cpu().numpy().transpose(1, 2, 0)
    if img_array.mean() > 1:
        img_array = (img_array / 255).clip(0, 1)
    else:
        img_array = img_array.clip(0, 1)
    plt.figure()
    plt.imshow(img_array)
    plt.title(title_str)
    plt.axis('off')


@torch.no_grad()
def show_interpolation(model, z_dim, d):
    # make 10x10 grids from interpolation within [-d, d] in latent space
    inter_range = torch.linspace(-d, d, 10)
    xs, ys = torch.meshgrid([inter_range, inter_range], indexing='xy')
    latents = torch.zeros(100, z_dim)
    latents[:, :2] = torch.stack([xs, ys], dim=-1).reshape(-1, 2)    #␣
 ↪interpolate on the first 2 dims
    samples = model.Generator(latents)
    samples = (samples + 1) / 2
    vis_grid = make_grid(samples, nrow=10)
    visualize_img_tensor(vis_grid, 'Interpolation in latent space')


def visualize_curve(losses, energy_pos_list, energy_neg_list):
    plt.figure()
    plt.subplot(1, 2, 1)
    plt.plot(losses)
    plt.title('Loss over iterations')
    plt.xlabel('Iteration')
    plt.ylabel('Loss')

    plt.subplot(1, 2, 2)
    plt.plot(energy_pos_list, label='positive')
```

```python
    plt.plot(energy_neg_list, label='negative')
    plt.title('Energy over iterations')
    plt.xlabel('Iteration')
    plt.ylabel('Energy')
    plt.legend()
    plt.tight_layout()


def visualize_samples(model, zs_post, imgs):
    z_random = torch.randn_like(zs_post[0])
    nearest_idx = torch.linalg.norm(zs_post-z_random, dim=-1).argmin()
    z_warm = zs_post[nearest_idx]
    I = imgs[nearest_idx]

    plt.figure()
    plt.subplot(1, 3, 1)
    plt.imshow(sample_img(model, z_random))
    plt.title('Random sample')
    plt.axis('off')

    plt.subplot(1, 3, 2)
    z_cold = model.sample_langevin_post(z_random, I)
    plt.imshow(sample_img(model, z_cold))
    plt.title('Posterior cold')
    plt.axis('off')

    plt.subplot(1, 3, 3)
    z_warm = model.sample_langevin_post(z_warm, I)
    plt.imshow(sample_img(model, z_warm))
    plt.title('Posterior warm')
    plt.axis('off')
    plt.tight_layout()
```

Experiment

For each setting z_dim = {2, 8, 32}, train the descriptor and generator by alternating backpropagation in warm start scheme. After finishing the epochs, show experimental results in three part:

- Visualize the curves of loss and positive/negative energy
- Display the synthesized images from random sample, cold start and warm start respectively
- Probe the interpolation in latent space, where we make 10x10 grids on the first 2 dimensions of latent space

[ ]:
```python
def run(args):
    imgs = load_data()
    print(f'Loaded {len(imgs)} images')
    z_dim = args.z_dim
```

```python
    zs_prior = torch.randn(len(imgs), z_dim)
    zs_post = torch.randn(len(imgs), z_dim)
    model = Model(args)
    energy_pos_list = []
    energy_neg_list = []
    losses = []
    for i in range(args.epochs):
        zs_prior, zs_post, energy_pos, energy_neg, loss = model.
 ↪update(zs_prior, zs_post, imgs)
        losses.append(loss)
        energy_pos_list.append(energy_pos)
        energy_neg_list.append(energy_neg)
        if (i+1) % 10 == 0:
            print(f'Epoch {i+1} loss: {loss:.4f}')

    visualize_curve(losses, energy_pos_list, energy_neg_list)

    visualize_samples(model, zs_post, imgs)

    show_interpolation(model, z_dim, args.interpolation_d)
```

z_dim = 2

[ ]:
```python
setattr(args, 'z_dim', 2)
run(args)
```

```
Loaded 11 images
Epoch 10 loss: 10645.5254
Epoch 20 loss: 10073.0157
Epoch 30 loss: 9583.7870
Epoch 40 loss: 9218.1420
Epoch 50 loss: 8888.1559
Epoch 60 loss: 8642.3575
Epoch 70 loss: 8403.2690
Epoch 80 loss: 8183.9230
Epoch 90 loss: 7962.8893
Epoch 100 loss: 7735.9093
Epoch 110 loss: 7524.6467
Epoch 120 loss: 7383.6735
Epoch 130 loss: 7120.1247
Epoch 140 loss: 6945.4719
Epoch 150 loss: 6719.0732
Epoch 160 loss: 6570.2642
Epoch 170 loss: 6345.5063
Epoch 180 loss: 6249.9855
Epoch 190 loss: 6124.1268
Epoch 200 loss: 5976.2718
Epoch 210 loss: 5784.8569
```

```
Epoch 220 loss: 5648.9274
Epoch 230 loss: 5539.6311
Epoch 240 loss: 5390.8978
Epoch 250 loss: 5258.7123
Epoch 260 loss: 5169.9640
Epoch 270 loss: 5002.8390
Epoch 280 loss: 4936.5046
Epoch 290 loss: 4810.5665
Epoch 300 loss: 4715.6262
Epoch 310 loss: 4697.1566
Epoch 320 loss: 4591.4579
Epoch 330 loss: 4510.5814
Epoch 340 loss: 4444.7629
Epoch 350 loss: 4410.0995
Epoch 360 loss: 4312.4395
Epoch 370 loss: 4274.0217
Epoch 380 loss: 4205.3011
Epoch 390 loss: 4124.5792
Epoch 400 loss: 4061.8546
Epoch 410 loss: 3993.6449
Epoch 420 loss: 3958.0993
Epoch 430 loss: 3968.1195
Epoch 440 loss: 3905.3684
Epoch 450 loss: 3907.1746
Epoch 460 loss: 3832.6833
Epoch 470 loss: 3752.8814
Epoch 480 loss: 3729.7213
Epoch 490 loss: 3685.3913
Epoch 500 loss: 3692.9826
Epoch 510 loss: 3719.4893
Epoch 520 loss: 3618.1321
Epoch 530 loss: 3609.7217
Epoch 540 loss: 3560.3698
Epoch 550 loss: 3514.7683
Epoch 560 loss: 3535.1716
Epoch 570 loss: 3468.2508
Epoch 580 loss: 3418.9756
Epoch 590 loss: 3380.6180
Epoch 600 loss: 3345.9588
Epoch 610 loss: 3343.7872
Epoch 620 loss: 3338.3636
Epoch 630 loss: 3311.9056
Epoch 640 loss: 3255.4236
Epoch 650 loss: 3293.7022
Epoch 660 loss: 3280.2220
Epoch 670 loss: 3213.4583
Epoch 680 loss: 3215.8859
Epoch 690 loss: 3212.1943
```

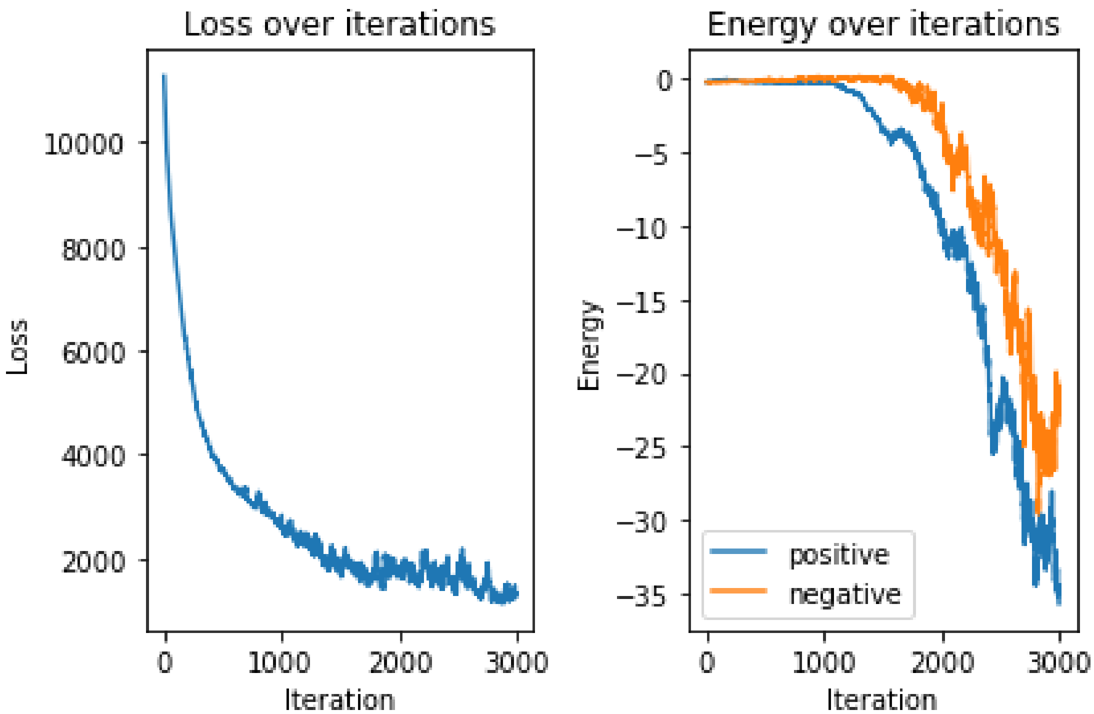
```
Epoch 700 loss: 3329.3246
Epoch 710 loss: 3170.8649
Epoch 720 loss: 3129.4132
Epoch 730 loss: 3097.7361
Epoch 740 loss: 3057.7296
Epoch 750 loss: 3059.7023
Epoch 760 loss: 3009.5130
Epoch 770 loss: 3028.0807
Epoch 780 loss: 2972.9222
Epoch 790 loss: 3113.1947
Epoch 800 loss: 3125.8471
Epoch 810 loss: 3194.6484
Epoch 820 loss: 3144.1845
Epoch 830 loss: 2974.2443
Epoch 840 loss: 2986.6601
Epoch 850 loss: 2881.3976
Epoch 860 loss: 2995.8364
Epoch 870 loss: 2895.9616
Epoch 880 loss: 2870.1637
Epoch 890 loss: 2923.9495
Epoch 900 loss: 2841.2027
Epoch 910 loss: 2844.5503
Epoch 920 loss: 2880.0785
Epoch 930 loss: 2823.0777
Epoch 940 loss: 2772.5286
Epoch 950 loss: 2764.3235
Epoch 960 loss: 2762.2195
Epoch 970 loss: 2677.7937
Epoch 980 loss: 2638.5174
Epoch 990 loss: 2689.4161
Epoch 1000 loss: 2572.2440
Epoch 1010 loss: 2598.0636
Epoch 1020 loss: 2601.8458
Epoch 1030 loss: 2533.4328
Epoch 1040 loss: 2493.3294
Epoch 1050 loss: 2444.1853
Epoch 1060 loss: 2555.6370
Epoch 1070 loss: 2687.5972
Epoch 1080 loss: 2721.8394
Epoch 1090 loss: 2496.5873
Epoch 1100 loss: 2519.7027
Epoch 1110 loss: 2409.5369
Epoch 1120 loss: 2363.1903
Epoch 1130 loss: 2253.2822
Epoch 1140 loss: 2250.5200
Epoch 1150 loss: 2259.1769
Epoch 1160 loss: 2317.3437
Epoch 1170 loss: 2456.0445
```

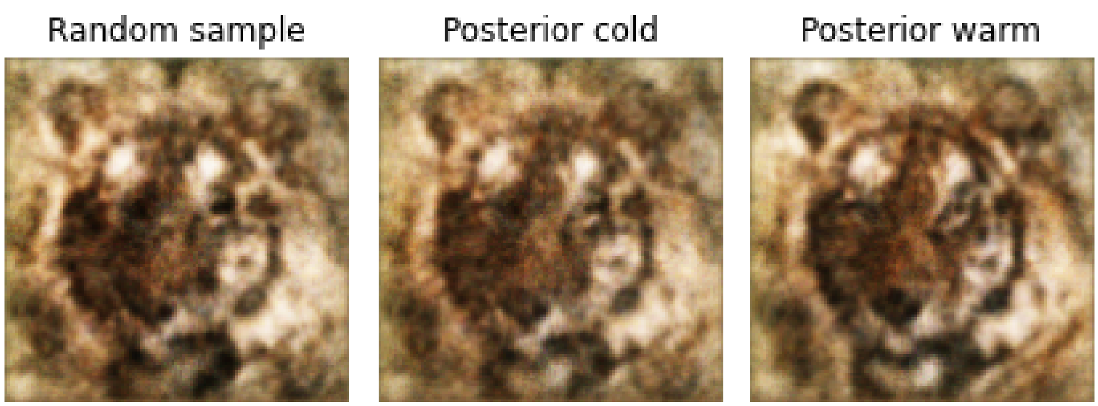
```
Epoch 1180 loss: 2454.7556
Epoch 1190 loss: 2401.1136
Epoch 1200 loss: 2337.3351
Epoch 1210 loss: 2260.0836
Epoch 1220 loss: 2313.3377
Epoch 1230 loss: 2356.7567
Epoch 1240 loss: 2415.1402
Epoch 1250 loss: 2196.2278
Epoch 1260 loss: 2224.0216
Epoch 1270 loss: 2091.0973
Epoch 1280 loss: 2077.4975
Epoch 1290 loss: 2455.8608
Epoch 1300 loss: 2265.2585
Epoch 1310 loss: 2225.7957
Epoch 1320 loss: 2190.3946
Epoch 1330 loss: 2113.3531
Epoch 1340 loss: 2174.7251
Epoch 1350 loss: 1997.2083
Epoch 1360 loss: 1910.0992
Epoch 1370 loss: 1879.7553
Epoch 1380 loss: 1982.7573
Epoch 1390 loss: 1878.8364
Epoch 1400 loss: 2022.1177
Epoch 1410 loss: 1909.7103
Epoch 1420 loss: 2027.8908
Epoch 1430 loss: 2151.3328
Epoch 1440 loss: 2013.9555
Epoch 1450 loss: 2060.6689
Epoch 1460 loss: 1811.2462
Epoch 1470 loss: 1876.1622
Epoch 1480 loss: 1772.8919
Epoch 1490 loss: 1881.0057
Epoch 1500 loss: 1749.8090
Epoch 1510 loss: 1713.1913
Epoch 1520 loss: 1893.3448
Epoch 1530 loss: 1924.9450
Epoch 1540 loss: 1931.7706
Epoch 1550 loss: 2072.2987
Epoch 1560 loss: 2006.2100
Epoch 1570 loss: 1961.0112
Epoch 1580 loss: 1879.2358
Epoch 1590 loss: 1821.0007
Epoch 1600 loss: 1837.7678
Epoch 1610 loss: 1933.9074
Epoch 1620 loss: 1752.1631
Epoch 1630 loss: 1765.9021
Epoch 1640 loss: 1835.4672
Epoch 1650 loss: 1725.9217
```

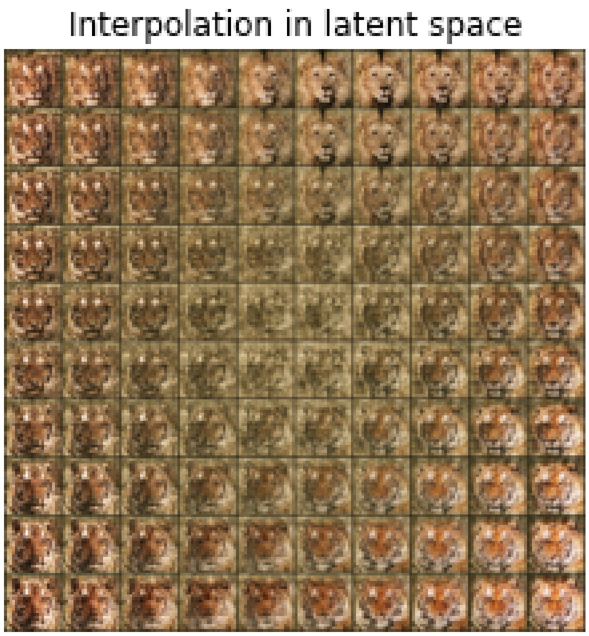
```
Epoch 1660 loss: 1671.5717
Epoch 1670 loss: 1569.9297
Epoch 1680 loss: 1850.4975
Epoch 1690 loss: 1718.5852
Epoch 1700 loss: 1742.2492
Epoch 1710 loss: 1697.0797
Epoch 1720 loss: 1707.8703
Epoch 1730 loss: 1502.4587
Epoch 1740 loss: 1600.7952
Epoch 1750 loss: 1509.4438
Epoch 1760 loss: 1667.9413
Epoch 1770 loss: 1613.4732
Epoch 1780 loss: 1651.4483
Epoch 1790 loss: 1741.5630
Epoch 1800 loss: 2080.3030
Epoch 1810 loss: 1964.8673
Epoch 1820 loss: 2017.0405
Epoch 1830 loss: 1626.0885
Epoch 1840 loss: 1523.5626
Epoch 1850 loss: 1496.9249
Epoch 1860 loss: 1792.1600
Epoch 1870 loss: 1439.5918
Epoch 1880 loss: 1605.9924
Epoch 1890 loss: 2000.7192
Epoch 1900 loss: 1825.8538
Epoch 1910 loss: 1705.2006
Epoch 1920 loss: 1640.2345
Epoch 1930 loss: 1882.4494
Epoch 1940 loss: 1904.0142
Epoch 1950 loss: 1985.4003
Epoch 1960 loss: 1797.8061
Epoch 1970 loss: 1798.5211
Epoch 1980 loss: 1728.0793
Epoch 1990 loss: 1634.7560
Epoch 2000 loss: 1733.5794
Epoch 2010 loss: 1759.0577
Epoch 2020 loss: 1778.9560
Epoch 2030 loss: 1806.0759
Epoch 2040 loss: 1959.4687
Epoch 2050 loss: 1896.7782
Epoch 2060 loss: 1952.2204
Epoch 2070 loss: 1842.9087
Epoch 2080 loss: 1905.3675
Epoch 2090 loss: 1635.8022
Epoch 2100 loss: 1798.4000
Epoch 2110 loss: 1869.7545
Epoch 2120 loss: 1773.8197
Epoch 2130 loss: 1662.8251
```

```
Epoch 2140 loss: 1706.4866
Epoch 2150 loss: 1547.2179
Epoch 2160 loss: 1538.1464
Epoch 2170 loss: 1555.7098
Epoch 2180 loss: 1543.1657
Epoch 2190 loss: 1545.4427
Epoch 2200 loss: 1942.1462
Epoch 2210 loss: 1977.0602
Epoch 2220 loss: 1853.9975
Epoch 2230 loss: 2134.3167
Epoch 2240 loss: 2136.1199
Epoch 2250 loss: 1791.8529
Epoch 2260 loss: 1813.3515
Epoch 2270 loss: 1707.6096
Epoch 2280 loss: 1539.1192
Epoch 2290 loss: 1577.8104
Epoch 2300 loss: 1774.7804
Epoch 2310 loss: 1872.7486
Epoch 2320 loss: 1818.2179
Epoch 2330 loss: 1525.5508
Epoch 2340 loss: 1746.4392
Epoch 2350 loss: 1694.5237
Epoch 2360 loss: 1509.7038
Epoch 2370 loss: 1453.2518
Epoch 2380 loss: 1718.4255
Epoch 2390 loss: 1728.4350
Epoch 2400 loss: 1754.4069
Epoch 2410 loss: 1863.3387
Epoch 2420 loss: 1765.6158
Epoch 2430 loss: 1593.8175
Epoch 2440 loss: 1579.8603
Epoch 2450 loss: 1624.0858
Epoch 2460 loss: 1431.6861
Epoch 2470 loss: 1432.4065
Epoch 2480 loss: 1340.8258
Epoch 2490 loss: 1376.2605
Epoch 2500 loss: 1672.9285
Epoch 2510 loss: 1908.1827
Epoch 2520 loss: 1748.8138
Epoch 2530 loss: 1902.4303
Epoch 2540 loss: 1975.5706
Epoch 2550 loss: 1762.5502
Epoch 2560 loss: 1541.9487
Epoch 2570 loss: 1623.4957
Epoch 2580 loss: 1607.0329
Epoch 2590 loss: 1615.6181
Epoch 2600 loss: 1823.9898
Epoch 2610 loss: 1595.5467
```

```
Epoch 2620 loss: 1445.5452
Epoch 2630 loss: 1563.0154
Epoch 2640 loss: 1458.2719
Epoch 2650 loss: 1432.4842
Epoch 2660 loss: 1326.6194
Epoch 2670 loss: 1316.6632
Epoch 2680 loss: 1325.3952
Epoch 2690 loss: 1305.2966
Epoch 2700 loss: 1259.6782
Epoch 2710 loss: 1242.4330
Epoch 2720 loss: 1360.4550
Epoch 2730 loss: 1507.8460
Epoch 2740 loss: 1450.2512
Epoch 2750 loss: 1671.5984
Epoch 2760 loss: 1892.2417
Epoch 2770 loss: 1543.2308
Epoch 2780 loss: 1427.7538
Epoch 2790 loss: 1181.8293
Epoch 2800 loss: 1218.6121
Epoch 2810 loss: 1347.0839
Epoch 2820 loss: 1254.0531
Epoch 2830 loss: 1364.4765
Epoch 2840 loss: 1158.3967
Epoch 2850 loss: 1308.6614
Epoch 2860 loss: 1189.7619
Epoch 2870 loss: 1203.4805
Epoch 2880 loss: 1281.0913
Epoch 2890 loss: 1215.7022
Epoch 2900 loss: 1298.6132
Epoch 2910 loss: 1343.3934
Epoch 2920 loss: 1466.1330
Epoch 2930 loss: 1389.3173
Epoch 2940 loss: 1276.1887
Epoch 2950 loss: 1375.6628
Epoch 2960 loss: 1374.7169
Epoch 2970 loss: 1251.3342
Epoch 2980 loss: 1250.1807
Epoch 2990 loss: 1442.1303
Epoch 3000 loss: 1298.2744
```


Loss over iterations
Loss
10000
8000
6000
4000
2000
0
1000
2000
3000
Iteration
Energy over iterations
Energy
0
−5
−10
−15
−20
−25
−30
−35
positive
negative
0
1000
2000
3000
Iteration


Random sample
Posterior cold
Posterior warm



z_dim = 8

[ ]:
```
setattr(args, 'z_dim', 8)
run(args)
```

```
Loaded 11 images
Epoch 10 loss: 10351.4913
Epoch 20 loss: 9789.6611
Epoch 30 loss: 9331.0901
Epoch 40 loss: 8965.5679
Epoch 50 loss: 8642.0639
Epoch 60 loss: 8384.2365
Epoch 70 loss: 8174.0254
Epoch 80 loss: 7966.3216
Epoch 90 loss: 7790.9810
Epoch 100 loss: 7618.4604
Epoch 110 loss: 7453.7207
Epoch 120 loss: 7310.3095
Epoch 130 loss: 7132.0281
Epoch 140 loss: 6976.0516
Epoch 150 loss: 6800.8930
Epoch 160 loss: 6637.1368
Epoch 170 loss: 6505.5512
Epoch 180 loss: 6351.1966
Epoch 190 loss: 6206.6371
Epoch 200 loss: 6071.8955
Epoch 210 loss: 5937.0407
```

```
Epoch 220 loss: 5804.9421
Epoch 230 loss: 5674.3318
Epoch 240 loss: 5539.9447
Epoch 250 loss: 5415.7275
Epoch 260 loss: 5306.7323
Epoch 270 loss: 5198.3362
Epoch 280 loss: 5077.7666
Epoch 290 loss: 4981.4128
Epoch 300 loss: 4901.3811
Epoch 310 loss: 4813.4779
Epoch 320 loss: 4715.1450
Epoch 330 loss: 4628.6244
Epoch 340 loss: 4556.9697
Epoch 350 loss: 4477.8588
Epoch 360 loss: 4418.5040
Epoch 370 loss: 4339.6450
Epoch 380 loss: 4279.8129
Epoch 390 loss: 4228.2565
Epoch 400 loss: 4180.5994
Epoch 410 loss: 4119.6439
Epoch 420 loss: 4075.3174
Epoch 430 loss: 4012.5001
Epoch 440 loss: 3978.7112
Epoch 450 loss: 3959.3792
Epoch 460 loss: 3921.1207
Epoch 470 loss: 3873.5344
Epoch 480 loss: 3829.1974
Epoch 490 loss: 3789.5306
Epoch 500 loss: 3744.8371
Epoch 510 loss: 3683.4886
Epoch 520 loss: 3635.0206
Epoch 530 loss: 3588.1720
Epoch 540 loss: 3543.4359
Epoch 550 loss: 3535.0463
Epoch 560 loss: 3493.9304
Epoch 570 loss: 3438.8972
Epoch 580 loss: 3378.1949
Epoch 590 loss: 3328.4922
Epoch 600 loss: 3274.3863
Epoch 610 loss: 3221.3888
Epoch 620 loss: 3148.7942
Epoch 630 loss: 3152.5124
Epoch 640 loss: 3073.9421
Epoch 650 loss: 2985.8500
Epoch 660 loss: 2949.5844
Epoch 670 loss: 2884.6939
Epoch 680 loss: 2810.2143
Epoch 690 loss: 2801.0784
```

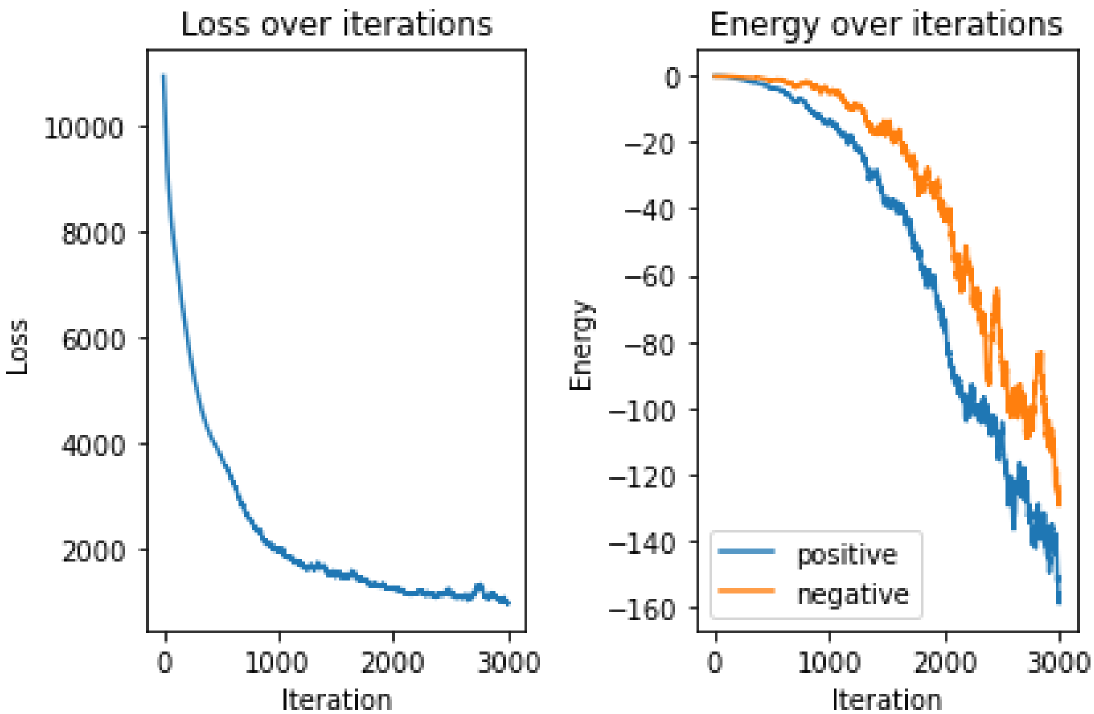
```
Epoch 700 loss: 2767.8085
Epoch 710 loss: 2697.2011
Epoch 720 loss: 2648.3528
Epoch 730 loss: 2610.3256
Epoch 740 loss: 2586.0649
Epoch 750 loss: 2539.6015
Epoch 760 loss: 2556.2455
Epoch 770 loss: 2468.5573
Epoch 780 loss: 2453.9001
Epoch 790 loss: 2404.5003
Epoch 800 loss: 2376.3405
Epoch 810 loss: 2335.5557
Epoch 820 loss: 2350.9977
Epoch 830 loss: 2291.3891
Epoch 840 loss: 2318.8450
Epoch 850 loss: 2302.0505
Epoch 860 loss: 2193.7531
Epoch 870 loss: 2137.6120
Epoch 880 loss: 2121.9310
Epoch 890 loss: 2136.6696
Epoch 900 loss: 2098.8707
Epoch 910 loss: 2078.0175
Epoch 920 loss: 2073.8819
Epoch 930 loss: 2067.9661
Epoch 940 loss: 2035.5125
Epoch 950 loss: 2025.9046
Epoch 960 loss: 1989.2106
Epoch 970 loss: 1978.9730
Epoch 980 loss: 1984.8848
Epoch 990 loss: 1989.2606
Epoch 1000 loss: 1963.4043
Epoch 1010 loss: 1972.5094
Epoch 1020 loss: 1984.4632
Epoch 1030 loss: 1935.2497
Epoch 1040 loss: 1970.5380
Epoch 1050 loss: 1872.0550
Epoch 1060 loss: 1861.5386
Epoch 1070 loss: 1846.1486
Epoch 1080 loss: 1829.9326
Epoch 1090 loss: 1824.5108
Epoch 1100 loss: 1833.4858
Epoch 1110 loss: 1800.9940
Epoch 1120 loss: 1821.7355
Epoch 1130 loss: 1752.8054
Epoch 1140 loss: 1732.7713
Epoch 1150 loss: 1761.9183
Epoch 1160 loss: 1750.7498
Epoch 1170 loss: 1714.6519
```

```
Epoch 1180 loss: 1695.9389
Epoch 1190 loss: 1734.1832
Epoch 1200 loss: 1704.0233
Epoch 1210 loss: 1655.3984
Epoch 1220 loss: 1656.6585
Epoch 1230 loss: 1637.2935
Epoch 1240 loss: 1617.6191
Epoch 1250 loss: 1632.5303
Epoch 1260 loss: 1614.6112
Epoch 1270 loss: 1657.4074
Epoch 1280 loss: 1653.8641
Epoch 1290 loss: 1653.2687
Epoch 1300 loss: 1629.7484
Epoch 1310 loss: 1595.7460
Epoch 1320 loss: 1639.8038
Epoch 1330 loss: 1637.3817
Epoch 1340 loss: 1712.2368
Epoch 1350 loss: 1684.6153
Epoch 1360 loss: 1650.4306
Epoch 1370 loss: 1613.9931
Epoch 1380 loss: 1646.8913
Epoch 1390 loss: 1629.5129
Epoch 1400 loss: 1632.1266
Epoch 1410 loss: 1618.5439
Epoch 1420 loss: 1540.6365
Epoch 1430 loss: 1519.6298
Epoch 1440 loss: 1478.6446
Epoch 1450 loss: 1512.9354
Epoch 1460 loss: 1530.0248
Epoch 1470 loss: 1508.4559
Epoch 1480 loss: 1541.4149
Epoch 1490 loss: 1525.4864
Epoch 1500 loss: 1462.8204
Epoch 1510 loss: 1494.1586
Epoch 1520 loss: 1487.6700
Epoch 1530 loss: 1544.6114
Epoch 1540 loss: 1492.0140
Epoch 1550 loss: 1449.8707
Epoch 1560 loss: 1488.1021
Epoch 1570 loss: 1485.8978
Epoch 1580 loss: 1482.0978
Epoch 1590 loss: 1479.0780
Epoch 1600 loss: 1467.1321
Epoch 1610 loss: 1470.3660
Epoch 1620 loss: 1468.5196
Epoch 1630 loss: 1487.6506
Epoch 1640 loss: 1541.6143
Epoch 1650 loss: 1491.6577
```

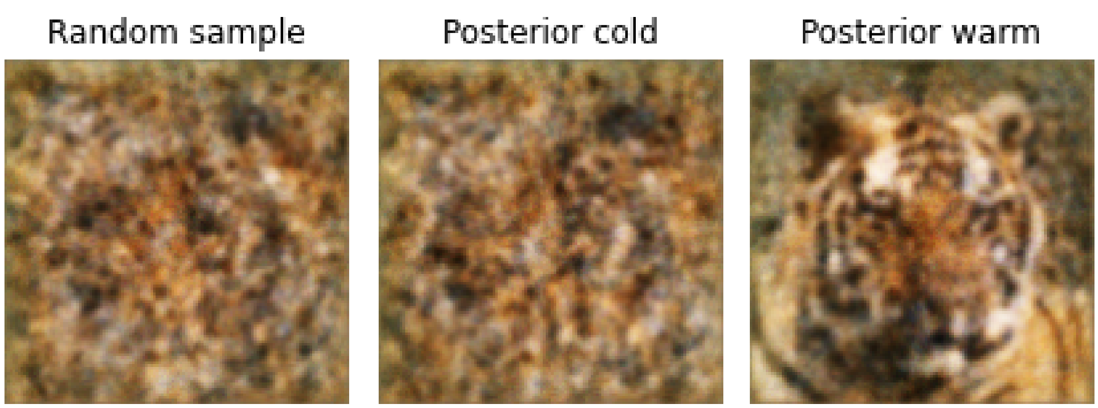
```
Epoch 1660 loss: 1524.7047
Epoch 1670 loss: 1525.9762
Epoch 1680 loss: 1454.6730
Epoch 1690 loss: 1460.5161
Epoch 1700 loss: 1389.4385
Epoch 1710 loss: 1425.1303
Epoch 1720 loss: 1432.5943
Epoch 1730 loss: 1431.3560
Epoch 1740 loss: 1402.0982
Epoch 1750 loss: 1389.6476
Epoch 1760 loss: 1399.9138
Epoch 1770 loss: 1365.1115
Epoch 1780 loss: 1346.8721
Epoch 1790 loss: 1311.4921
Epoch 1800 loss: 1318.6047
Epoch 1810 loss: 1314.1340
Epoch 1820 loss: 1304.0950
Epoch 1830 loss: 1310.6264
Epoch 1840 loss: 1321.8632
Epoch 1850 loss: 1320.6927
Epoch 1860 loss: 1306.3157
Epoch 1870 loss: 1313.3850
Epoch 1880 loss: 1268.5966
Epoch 1890 loss: 1277.7806
Epoch 1900 loss: 1286.8169
Epoch 1910 loss: 1315.6047
Epoch 1920 loss: 1305.6672
Epoch 1930 loss: 1251.4703
Epoch 1940 loss: 1237.0323
Epoch 1950 loss: 1237.2529
Epoch 1960 loss: 1252.0023
Epoch 1970 loss: 1249.2850
Epoch 1980 loss: 1223.7177
Epoch 1990 loss: 1241.7233
Epoch 2000 loss: 1249.9561
Epoch 2010 loss: 1233.5805
Epoch 2020 loss: 1211.2918
Epoch 2030 loss: 1228.7727
Epoch 2040 loss: 1217.2600
Epoch 2050 loss: 1212.9834
Epoch 2060 loss: 1188.6810
Epoch 2070 loss: 1184.6723
Epoch 2080 loss: 1213.3321
Epoch 2090 loss: 1172.6800
Epoch 2100 loss: 1156.3177
Epoch 2110 loss: 1152.3150
Epoch 2120 loss: 1139.5519
Epoch 2130 loss: 1144.6039
```

```
Epoch 2140 loss: 1133.3359
Epoch 2150 loss: 1125.9907
Epoch 2160 loss: 1128.7789
Epoch 2170 loss: 1135.7114
Epoch 2180 loss: 1140.4798
Epoch 2190 loss: 1159.8899
Epoch 2200 loss: 1138.3624
Epoch 2210 loss: 1168.7694
Epoch 2220 loss: 1182.7483
Epoch 2230 loss: 1201.0269
Epoch 2240 loss: 1175.4195
Epoch 2250 loss: 1153.1507
Epoch 2260 loss: 1160.9686
Epoch 2270 loss: 1127.6628
Epoch 2280 loss: 1118.4534
Epoch 2290 loss: 1157.3855
Epoch 2300 loss: 1144.9563
Epoch 2310 loss: 1105.4877
Epoch 2320 loss: 1113.7561
Epoch 2330 loss: 1126.7126
Epoch 2340 loss: 1129.9889
Epoch 2350 loss: 1114.7235
Epoch 2360 loss: 1103.1710
Epoch 2370 loss: 1111.4494
Epoch 2380 loss: 1085.2792
Epoch 2390 loss: 1081.5640
Epoch 2400 loss: 1111.2800
Epoch 2410 loss: 1125.7629
Epoch 2420 loss: 1117.4076
Epoch 2430 loss: 1142.7838
Epoch 2440 loss: 1119.4874
Epoch 2450 loss: 1163.5300
Epoch 2460 loss: 1159.8394
Epoch 2470 loss: 1179.3978
Epoch 2480 loss: 1186.5961
Epoch 2490 loss: 1138.1976
Epoch 2500 loss: 1107.7907
Epoch 2510 loss: 1136.0697
Epoch 2520 loss: 1152.3564
Epoch 2530 loss: 1089.2346
Epoch 2540 loss: 1095.9815
Epoch 2550 loss: 1067.2260
Epoch 2560 loss: 1078.9954
Epoch 2570 loss: 1066.6039
Epoch 2580 loss: 1091.2646
Epoch 2590 loss: 1074.9313
Epoch 2600 loss: 1065.9445
Epoch 2610 loss: 1073.0881
```

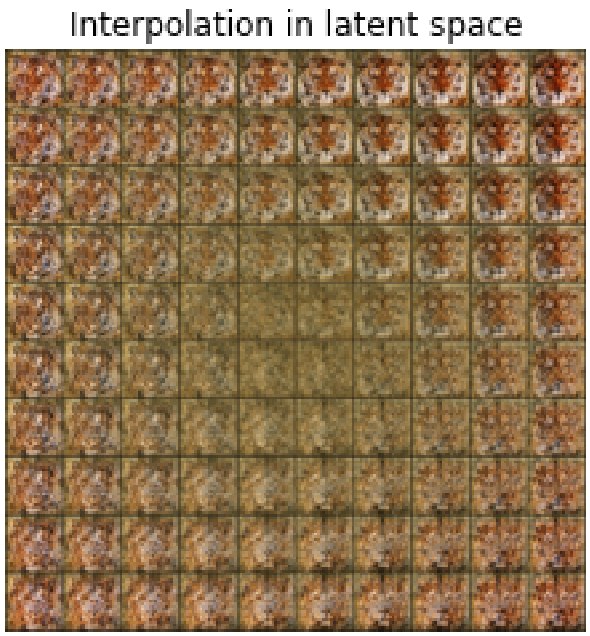
```
Epoch 2620 loss: 1097.4942
Epoch 2630 loss: 1104.7305
Epoch 2640 loss: 1082.3922
Epoch 2650 loss: 1049.5048
Epoch 2660 loss: 1081.5473
Epoch 2670 loss: 1079.5754
Epoch 2680 loss: 1077.1895
Epoch 2690 loss: 1064.0351
Epoch 2700 loss: 1141.9452
Epoch 2710 loss: 1140.5111
Epoch 2720 loss: 1191.9500
Epoch 2730 loss: 1212.0777
Epoch 2740 loss: 1256.3554
Epoch 2750 loss: 1255.3432
Epoch 2760 loss: 1312.4536
Epoch 2770 loss: 1232.4733
Epoch 2780 loss: 1222.7242
Epoch 2790 loss: 1138.9648
Epoch 2800 loss: 1087.4574
Epoch 2810 loss: 1076.8515
Epoch 2820 loss: 1061.1357
Epoch 2830 loss: 1069.5621
Epoch 2840 loss: 1066.8444
Epoch 2850 loss: 1130.3021
Epoch 2860 loss: 1149.2296
Epoch 2870 loss: 1108.1640
Epoch 2880 loss: 1091.9809
Epoch 2890 loss: 1078.2623
Epoch 2900 loss: 1066.5389
Epoch 2910 loss: 1040.9116
Epoch 2920 loss: 1029.0232
Epoch 2930 loss: 1027.8684
Epoch 2940 loss: 1007.9404
Epoch 2950 loss: 1007.3565
Epoch 2960 loss: 1009.9366
Epoch 2970 loss: 1033.1152
Epoch 2980 loss: 994.0114
Epoch 2990 loss: 964.0683
Epoch 3000 loss: 959.7170
```



Loss over iterations
Loss
10000
8000
6000
4000
2000
0
1000
2000
3000
Iteration
Energy over iterations
Energy
0
−20
−40
−60
−80
−100
−120
−140
−160
0
1000
2000
3000
Iteration
positive
negative



Random sample
Posterior cold
Posterior warm



z_dim = 32

```
[ ]: setattr(args, 'z_dim', 32)
run(args)
```

```
Loaded 11 images
Epoch 10 loss: 10264.5957
Epoch 20 loss: 9719.8509
Epoch 30 loss: 9258.0228
Epoch 40 loss: 8897.4096
Epoch 50 loss: 8641.4690
Epoch 60 loss: 8412.2803
Epoch 70 loss: 8228.8242
Epoch 80 loss: 8053.0207
Epoch 90 loss: 7873.2373
Epoch 100 loss: 7690.3029
Epoch 110 loss: 7521.9621
Epoch 120 loss: 7352.3430
Epoch 130 loss: 7187.2573
Epoch 140 loss: 7041.3908
Epoch 150 loss: 6877.6409
Epoch 160 loss: 6718.0569
Epoch 170 loss: 6569.6394
Epoch 180 loss: 6431.7974
Epoch 190 loss: 6289.1866
Epoch 200 loss: 6152.7978
Epoch 210 loss: 6018.4960
```

```
Epoch 220 loss: 5885.1405
Epoch 230 loss: 5767.2153
Epoch 240 loss: 5637.2656
Epoch 250 loss: 5512.9186
Epoch 260 loss: 5406.7713
Epoch 270 loss: 5298.8986
Epoch 280 loss: 5187.6329
Epoch 290 loss: 5090.7018
Epoch 300 loss: 4992.7543
Epoch 310 loss: 4907.5509
Epoch 320 loss: 4828.1025
Epoch 330 loss: 4750.3995
Epoch 340 loss: 4673.2170
Epoch 350 loss: 4598.0694
Epoch 360 loss: 4527.0781
Epoch 370 loss: 4459.3821
Epoch 380 loss: 4397.9230
Epoch 390 loss: 4340.8250
Epoch 400 loss: 4288.2878
Epoch 410 loss: 4243.2263
Epoch 420 loss: 4193.1793
Epoch 430 loss: 4147.8832
Epoch 440 loss: 4108.6855
Epoch 450 loss: 4071.0292
Epoch 460 loss: 4033.9172
Epoch 470 loss: 4002.7092
Epoch 480 loss: 3965.3816
Epoch 490 loss: 3939.4888
Epoch 500 loss: 3914.1645
Epoch 510 loss: 3890.4148
Epoch 520 loss: 3858.9216
Epoch 530 loss: 3830.0218
Epoch 540 loss: 3804.7661
Epoch 550 loss: 3785.4467
Epoch 560 loss: 3769.9258
Epoch 570 loss: 3745.3689
Epoch 580 loss: 3729.0206
Epoch 590 loss: 3713.6035
Epoch 600 loss: 3684.6905
Epoch 610 loss: 3664.5883
Epoch 620 loss: 3646.9027
Epoch 630 loss: 3623.9237
Epoch 640 loss: 3605.4547
Epoch 650 loss: 3583.6470
Epoch 660 loss: 3563.6483
Epoch 670 loss: 3538.2379
Epoch 680 loss: 3516.3066
Epoch 690 loss: 3488.4340
```

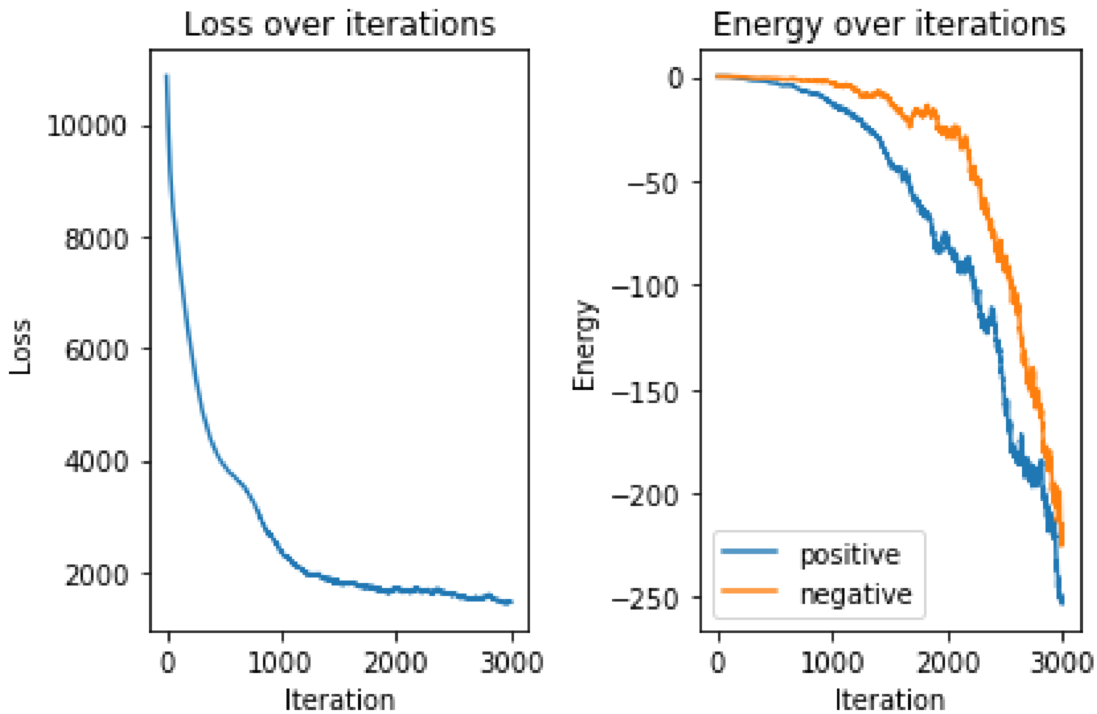
```
Epoch 700 loss: 3451.4322
Epoch 710 loss: 3425.7235
Epoch 720 loss: 3388.4223
Epoch 730 loss: 3357.9442
Epoch 740 loss: 3325.8916
Epoch 750 loss: 3281.7560
Epoch 760 loss: 3244.2998
Epoch 770 loss: 3195.9489
Epoch 780 loss: 3151.7383
Epoch 790 loss: 3099.3609
Epoch 800 loss: 3076.7604
Epoch 810 loss: 2997.5233
Epoch 820 loss: 2975.2884
Epoch 830 loss: 2913.0858
Epoch 840 loss: 2859.9924
Epoch 850 loss: 2832.9185
Epoch 860 loss: 2796.7292
Epoch 870 loss: 2749.8599
Epoch 880 loss: 2721.6550
Epoch 890 loss: 2688.0914
Epoch 900 loss: 2657.7898
Epoch 910 loss: 2654.7339
Epoch 920 loss: 2624.4132
Epoch 930 loss: 2591.0434
Epoch 940 loss: 2548.7184
Epoch 950 loss: 2539.0042
Epoch 960 loss: 2519.9998
Epoch 970 loss: 2459.0515
Epoch 980 loss: 2420.3981
Epoch 990 loss: 2379.9182
Epoch 1000 loss: 2358.7036
Epoch 1010 loss: 2344.9964
Epoch 1020 loss: 2319.2590
Epoch 1030 loss: 2270.7317
Epoch 1040 loss: 2263.1924
Epoch 1050 loss: 2277.8067
Epoch 1060 loss: 2256.6574
Epoch 1070 loss: 2239.2929
Epoch 1080 loss: 2209.1173
Epoch 1090 loss: 2180.0176
Epoch 1100 loss: 2160.5778
Epoch 1110 loss: 2145.6366
Epoch 1120 loss: 2125.2664
Epoch 1130 loss: 2126.3724
Epoch 1140 loss: 2078.6235
Epoch 1150 loss: 2084.8009
Epoch 1160 loss: 2084.6936
Epoch 1170 loss: 2065.8859
```

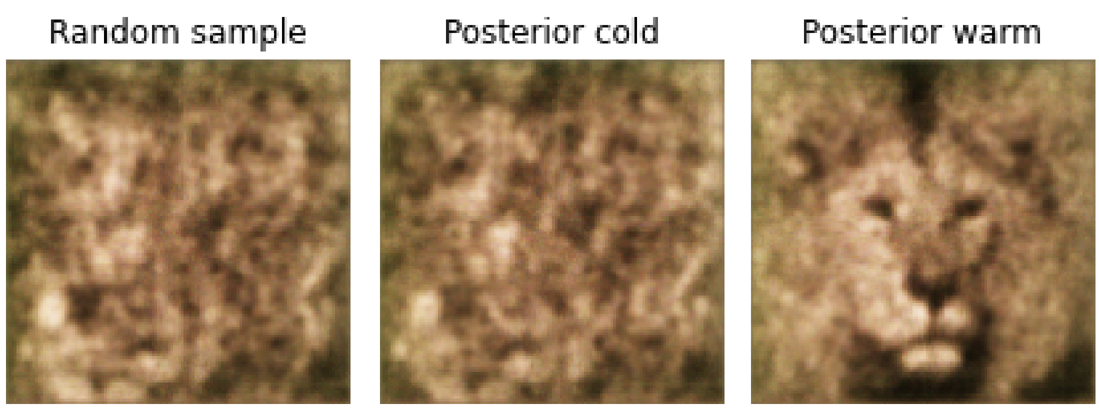
```
Epoch 1180 loss: 2040.5013
Epoch 1190 loss: 2046.6028
Epoch 1200 loss: 2018.1513
Epoch 1210 loss: 1989.4999
Epoch 1220 loss: 1983.0938
Epoch 1230 loss: 1971.7512
Epoch 1240 loss: 1958.6361
Epoch 1250 loss: 1967.1547
Epoch 1260 loss: 1971.4827
Epoch 1270 loss: 1952.7296
Epoch 1280 loss: 1960.0714
Epoch 1290 loss: 1964.1957
Epoch 1300 loss: 1949.0060
Epoch 1310 loss: 1961.9166
Epoch 1320 loss: 1970.3460
Epoch 1330 loss: 1958.5880
Epoch 1340 loss: 1933.1703
Epoch 1350 loss: 1909.1044
Epoch 1360 loss: 1916.0248
Epoch 1370 loss: 1896.3294
Epoch 1380 loss: 1885.6630
Epoch 1390 loss: 1902.9215
Epoch 1400 loss: 1869.9799
Epoch 1410 loss: 1855.3013
Epoch 1420 loss: 1872.2001
Epoch 1430 loss: 1894.8081
Epoch 1440 loss: 1853.9500
Epoch 1450 loss: 1840.3582
Epoch 1460 loss: 1815.6326
Epoch 1470 loss: 1812.8858
Epoch 1480 loss: 1845.9905
Epoch 1490 loss: 1851.1514
Epoch 1500 loss: 1806.7516
Epoch 1510 loss: 1806.9365
Epoch 1520 loss: 1789.9929
Epoch 1530 loss: 1799.3168
Epoch 1540 loss: 1800.6150
Epoch 1550 loss: 1795.8125
Epoch 1560 loss: 1807.3725
Epoch 1570 loss: 1809.6671
Epoch 1580 loss: 1799.4844
Epoch 1590 loss: 1797.5496
Epoch 1600 loss: 1808.1909
Epoch 1610 loss: 1815.5487
Epoch 1620 loss: 1812.4745
Epoch 1630 loss: 1815.6339
Epoch 1640 loss: 1787.3017
Epoch 1650 loss: 1790.6154
```

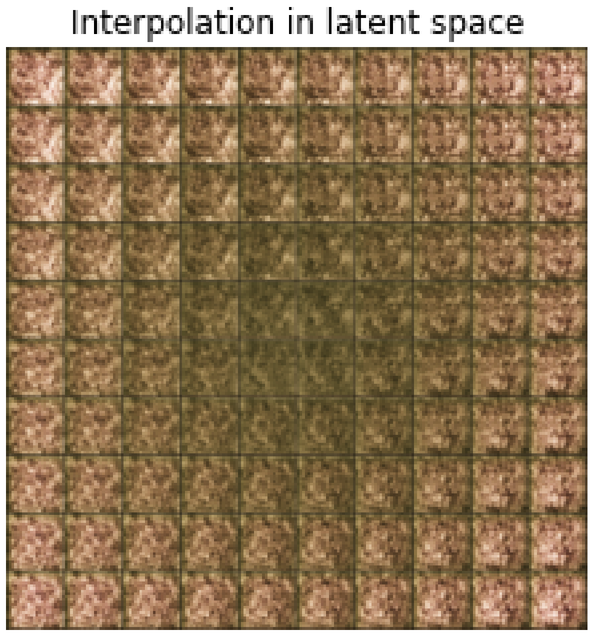
```
Epoch 1660 loss: 1767.5932
Epoch 1670 loss: 1770.1015
Epoch 1680 loss: 1747.7772
Epoch 1690 loss: 1743.6319
Epoch 1700 loss: 1743.3593
Epoch 1710 loss: 1741.5981
Epoch 1720 loss: 1727.3504
Epoch 1730 loss: 1754.4021
Epoch 1740 loss: 1752.1664
Epoch 1750 loss: 1738.1631
Epoch 1760 loss: 1746.3258
Epoch 1770 loss: 1728.9755
Epoch 1780 loss: 1734.4546
Epoch 1790 loss: 1735.7900
Epoch 1800 loss: 1727.2751
Epoch 1810 loss: 1721.6929
Epoch 1820 loss: 1705.8217
Epoch 1830 loss: 1690.8599
Epoch 1840 loss: 1690.6079
Epoch 1850 loss: 1706.8760
Epoch 1860 loss: 1705.1141
Epoch 1870 loss: 1672.0180
Epoch 1880 loss: 1676.0951
Epoch 1890 loss: 1674.0083
Epoch 1900 loss: 1673.6586
Epoch 1910 loss: 1655.4061
Epoch 1920 loss: 1642.2722
Epoch 1930 loss: 1648.5496
Epoch 1940 loss: 1645.2580
Epoch 1950 loss: 1648.3612
Epoch 1960 loss: 1656.9238
Epoch 1970 loss: 1696.4054
Epoch 1980 loss: 1683.3691
Epoch 1990 loss: 1677.6976
Epoch 2000 loss: 1705.5141
Epoch 2010 loss: 1716.8531
Epoch 2020 loss: 1697.7185
Epoch 2030 loss: 1695.3169
Epoch 2040 loss: 1688.1986
Epoch 2050 loss: 1662.4129
Epoch 2060 loss: 1654.8913
Epoch 2070 loss: 1670.3630
Epoch 2080 loss: 1669.5690
Epoch 2090 loss: 1667.3584
Epoch 2100 loss: 1659.7343
Epoch 2110 loss: 1657.5779
Epoch 2120 loss: 1639.2346
Epoch 2130 loss: 1649.0106
```

```
Epoch 2140 loss: 1678.9289
Epoch 2150 loss: 1677.2276
Epoch 2160 loss: 1685.9554
Epoch 2170 loss: 1694.4539
Epoch 2180 loss: 1703.9715
Epoch 2190 loss: 1699.8930
Epoch 2200 loss: 1680.9828
Epoch 2210 loss: 1674.8828
Epoch 2220 loss: 1655.4990
Epoch 2230 loss: 1650.9779
Epoch 2240 loss: 1649.0469
Epoch 2250 loss: 1669.6299
Epoch 2260 loss: 1674.0798
Epoch 2270 loss: 1673.8705
Epoch 2280 loss: 1677.6962
Epoch 2290 loss: 1659.2507
Epoch 2300 loss: 1633.3642
Epoch 2310 loss: 1611.2748
Epoch 2320 loss: 1650.0237
Epoch 2330 loss: 1633.1806
Epoch 2340 loss: 1672.1206
Epoch 2350 loss: 1671.0340
Epoch 2360 loss: 1695.5036
Epoch 2370 loss: 1658.4716
Epoch 2380 loss: 1665.1593
Epoch 2390 loss: 1643.6706
Epoch 2400 loss: 1629.7539
Epoch 2410 loss: 1643.1591
Epoch 2420 loss: 1637.2078
Epoch 2430 loss: 1619.5062
Epoch 2440 loss: 1624.3199
Epoch 2450 loss: 1615.9186
Epoch 2460 loss: 1631.4647
Epoch 2470 loss: 1611.6069
Epoch 2480 loss: 1598.1762
Epoch 2490 loss: 1589.1577
Epoch 2500 loss: 1608.1139
Epoch 2510 loss: 1623.2950
Epoch 2520 loss: 1599.2485
Epoch 2530 loss: 1583.5176
Epoch 2540 loss: 1570.1504
Epoch 2550 loss: 1568.1418
Epoch 2560 loss: 1553.5095
Epoch 2570 loss: 1569.1391
Epoch 2580 loss: 1554.8998
Epoch 2590 loss: 1547.2145
Epoch 2600 loss: 1556.0109
Epoch 2610 loss: 1547.6379
```

```
Epoch 2620 loss: 1526.1295
Epoch 2630 loss: 1525.5835
Epoch 2640 loss: 1527.5630
Epoch 2650 loss: 1534.2352
Epoch 2660 loss: 1524.9600
Epoch 2670 loss: 1528.1825
Epoch 2680 loss: 1526.5622
Epoch 2690 loss: 1543.1289
Epoch 2700 loss: 1550.8299
Epoch 2710 loss: 1532.9563
Epoch 2720 loss: 1514.7867
Epoch 2730 loss: 1541.8601
Epoch 2740 loss: 1527.0474
Epoch 2750 loss: 1550.2342
Epoch 2760 loss: 1533.8828
Epoch 2770 loss: 1548.6369
Epoch 2780 loss: 1559.9320
Epoch 2790 loss: 1589.7261
Epoch 2800 loss: 1562.5223
Epoch 2810 loss: 1582.6946
Epoch 2820 loss: 1573.0306
Epoch 2830 loss: 1548.2866
Epoch 2840 loss: 1553.7883
Epoch 2850 loss: 1533.6342
Epoch 2860 loss: 1528.4049
Epoch 2870 loss: 1514.0458
Epoch 2880 loss: 1487.9169
Epoch 2890 loss: 1499.6063
Epoch 2900 loss: 1487.1657
Epoch 2910 loss: 1477.2224
Epoch 2920 loss: 1461.0386
Epoch 2930 loss: 1466.0431
Epoch 2940 loss: 1462.7264
Epoch 2950 loss: 1459.6572
Epoch 2960 loss: 1452.6845
Epoch 2970 loss: 1439.1975
Epoch 2980 loss: 1461.0488
Epoch 2990 loss: 1479.0107
Epoch 3000 loss: 1476.3254
```


Loss over iterations
Loss
10000
8000
6000
4000
2000
0
1000
2000
3000
Iteration
Energy over iterations
Energy
0
−50
−100
−150
−200
−250
positive
negative
0
1000
2000
3000
Iteration


Random sample
Posterior cold
Posterior warm


Interpolation in latent space