3DTerrianLSGAN_sgd

May 1, 2020

[41]:
```
%matplotlib inline
```

[42]:
```
from __future__ import print_function
%matplotlib inline
import argparse
import os
import random
import torch
import torch.nn as nn
import torch.nn.parallel
import torch.backends.cudnn as cudnn
import torch.optim as optim
import torch.utils.data
import torchvision.datasets as dset
import torchvision.transforms as transforms
import torchvision.utils as vutils
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.animation as animation
from IPython.display import HTML
from torchvision.utils import save_image
from pathlib import Path

# Set random seed for reproducibility
#manualSeed = 999
manualSeed = random.randint(1, 10000) # use if you want new results
print("Random Seed: ", manualSeed)
random.seed(manualSeed)
torch.manual_seed(manualSeed)
```

```
Random Seed:  3444
```

[42]:
```
<torch._C.Generator at 0x7f71084aaaf0>
```

[43]:
```
# Root directory for dataset
#dataroot = "C:/Users/yyhh/Desktop/Morpheus/data/celeba"
dataroot = "data/HTile128"
# Number of workers for dataloader
```

```
workers = 2

# Batch size during training
batch_size = 16

# Spatial size of training images. All images will be resized to this
#   size using a transformer.
image_size = 128

# Number of channels in the training images. For color images this is 3
nc = 3

# Size of z latent vector (i.e. size of generator input)
nz = 100

# Size of feature maps in generator
ngf = 128

# Size of feature maps in discriminator
ndf = 32

# Number of training epochs
num_epochs = 150

# Learning rate for optimizers
lr = 0.01

# Beta1 hyperparam for Adam optimizers
beta1 = 0.5

# Number of GPUs available. Use 0 for CPU mode.
ngpu = 1
```

[44]:

```
#create output
directoryName = "output/sgd_" + str(lr) + "_"+str(batch_size)
Path(directoryName).mkdir(parents=True, exist_ok=True)
```

[45]:

```
# We can use an image folder dataset the way we have it setup.
# Create the dataset
dataset = dset.ImageFolder(root=dataroot,
                           transform=transforms.Compose([
                               transforms.Resize(image_size),
                               transforms.CenterCrop(image_size),
                               #transforms.Grayscale(num_output_channels=1),
                               transforms.ToTensor(),
```

```
                               transforms.Normalize((0.5, 0.5, 0.5), (0.5, 0.5,␣
↪0.5)),
                               #transforms.Normalize((0.5,), (0.5,)),

                           ]))
# Create the dataloader
dataloader = torch.utils.data.DataLoader(dataset, batch_size=batch_size,
                                         shuffle=True, num_workers=workers)
#print(len(dataloader))
#print(dataloader)
#nextitem= next(iter(dataloader))
#print(type(dataloader))
#print(type(next(iter(dataloader))))
#print(len(next(iter(dataloader))))
#print(nextitem)
# Decide which device we want to run on
device = torch.device("cuda:0" if (torch.cuda.is_available() and ngpu > 0) else␣
↪"cpu")

print(torch.cuda.is_available())

# Plot some training images
real_batch = next(iter(dataloader))

print(type(real_batch))

#plt.figure(figsize=(8,8))
#plt.axis("off")
#plt.title("Training Images")
#plt.imshow(np.transpose(vutils.make_grid(real_batch[0].to(device)[:64],␣
↪padding=2, normalize=True).cpu(),(1,2,0)))
```

```
True
<class 'list'>
```

[46]:

```
real_batch = next(iter(dataloader))
print(type(real_batch))
print(type(real_batch[0]))
print(len(real_batch[0][0][0]))
#print(real_batch[0])
```

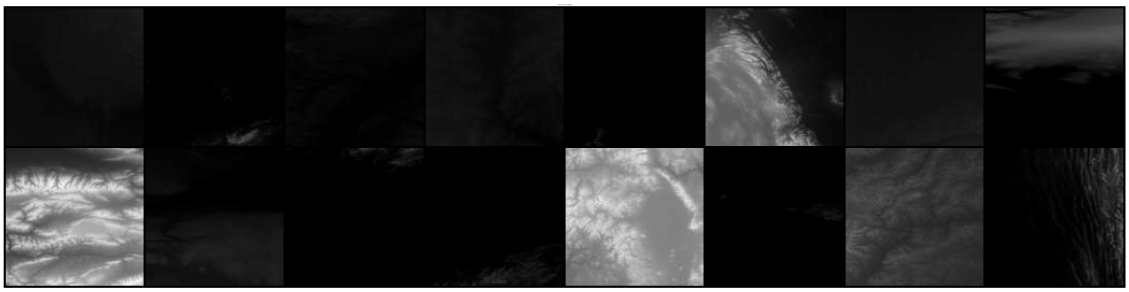
```
<class 'list'>
<class 'torch.Tensor'>
128
```

[47]:

```
plt.figure(figsize=(128,128))
plt.axis("off")
```

```
plt.title("Training Images")
plt.imshow(np.transpose(vutils.make_grid(real_batch[0].to(device)[:64],␣
↪padding=2, normalize=True).cpu(),(1,2,0)))
```

[47]: <matplotlib.image.AxesImage at 0x7f7082fe7fd0>



[48]:
```
# custom weights initialization called on netG and netD
def weights_init(m):
    classname = m.__class__.__name__
    if classname.find('Conv') != -1:
        nn.init.normal_(m.weight.data, 0.0, 0.02)
    elif classname.find('BatchNorm') != -1:
        nn.init.normal_(m.weight.data, 1.0, 0.02)
        nn.init.constant_(m.bias.data, 0)
```

[49]:
```
class Generator(nn.Module):
    def __init__(self, npgu):
        super(Generator, self).__init__()
        self.ngpu = ngpu
        self.init_size = 128 // 4
        self.l1 = nn.Sequential(nn.Linear(100, 128 * self.init_size ** 2))

        self.conv_blocks = nn.Sequential(
            nn.Upsample(scale_factor=2),
            nn.Conv2d(128, 128, 3, stride=1, padding=1),
            nn.BatchNorm2d(128, 0.8),
            nn.LeakyReLU(0.2, inplace=True),
            nn.Upsample(scale_factor=2),
            nn.Conv2d(128, 64, 3, stride=1, padding=1),
            nn.BatchNorm2d(64, 0.8),
            nn.LeakyReLU(0.2, inplace=True),
            nn.Conv2d(64, 3, 3, stride=1, padding=1),
            nn.Tanh(),
        )

    def forward(self, z):
```

```
        out = self.l1(z)
        out = out.view(out.shape[0], 128, self.init_size, self.init_size)
        img = self.conv_blocks(out)
        return img
```

[50]:
```
# Create the generator
netG = Generator(ngpu).to(device)

# Handle multi-gpu if desired
if (device.type == 'cuda') and (ngpu > 1):
    netG = nn.DataParallel(netG, list(range(ngpu)))

# Apply the weights_init function to randomly initialize all weights
#  to mean=0, stdev=0.2.
netG.apply(weights_init)

# Print the model
print(netG)

#from torchsummary import summary
#summary(netG, input_size= (100, 1024, 4))
```

```
Generator(
  (l1): Sequential(
    (0): Linear(in_features=100, out_features=131072, bias=True)
  )
  (conv_blocks): Sequential(
    (0): Upsample(scale_factor=2.0, mode=nearest)
    (1): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (2): BatchNorm2d(128, eps=0.8, momentum=0.1, affine=True,
track_running_stats=True)
    (3): LeakyReLU(negative_slope=0.2, inplace=True)
    (4): Upsample(scale_factor=2.0, mode=nearest)
    (5): Conv2d(128, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (6): BatchNorm2d(64, eps=0.8, momentum=0.1, affine=True,
track_running_stats=True)
    (7): LeakyReLU(negative_slope=0.2, inplace=True)
    (8): Conv2d(64, 3, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (9): Tanh()
  )
)
```

Discriminator Code

[51]:
```
class Discriminator(nn.Module):
    def __init__(self, ngpu):
        super(Discriminator, self).__init__()
        self.ngpu = ngpu
```

```
        def discriminator_block(in_filters, out_filters, bn=True):
            block = [nn.Conv2d(in_filters, out_filters, 3, 2, 1), nn.
↪LeakyReLU(0.2, inplace=True), nn.Dropout2d(0.25)]
            if bn:
                block.append(nn.BatchNorm2d(out_filters, 0.8))
            return block

        self.model = nn.Sequential(
            *discriminator_block(3, 16, bn=False),
            *discriminator_block(16, 32),
            *discriminator_block(32, 64),
            *discriminator_block(64, 128),
        )

        # The height and width of downsampled image
        ds_size = 128 // 2 ** 4
        self.adv_layer = nn.Linear(128 * ds_size ** 2, 1)

    def forward(self, img):
        out = self.model(img)
        out = out.view(out.shape[0], -1)
        validity = self.adv_layer(out)

        return validity
```

Now, as with the generator, we can create the discriminator, apply the `weights_init` function, and print the model's structure.

[52]:
```
# Create the Discriminator
netD = Discriminator(ngpu).to(device)

# Handle multi-gpu if desired
if (device.type == 'cuda') and (ngpu > 1):
    netD = nn.DataParallel(netD, list(range(ngpu)))

# Apply the weights_init function to randomly initialize all weights
#  to mean=0, stdev=0.2.
netD.apply(weights_init)

# Print the model
print(netD)

#from torchsummary import summary
#summary(netD, input_size = (3, 128, 128))
```

```
Discriminator(
  (model): Sequential(
```

```
    (0): Conv2d(3, 16, kernel_size=(3, 3), stride=(2, 2), padding=(1, 1))
    (1): LeakyReLU(negative_slope=0.2, inplace=True)
    (2): Dropout2d(p=0.25, inplace=False)
    (3): Conv2d(16, 32, kernel_size=(3, 3), stride=(2, 2), padding=(1, 1))
    (4): LeakyReLU(negative_slope=0.2, inplace=True)
    (5): Dropout2d(p=0.25, inplace=False)
    (6): BatchNorm2d(32, eps=0.8, momentum=0.1, affine=True,
track_running_stats=True)
    (7): Conv2d(32, 64, kernel_size=(3, 3), stride=(2, 2), padding=(1, 1))
    (8): LeakyReLU(negative_slope=0.2, inplace=True)
    (9): Dropout2d(p=0.25, inplace=False)
    (10): BatchNorm2d(64, eps=0.8, momentum=0.1, affine=True,
track_running_stats=True)
    (11): Conv2d(64, 128, kernel_size=(3, 3), stride=(2, 2), padding=(1, 1))
    (12): LeakyReLU(negative_slope=0.2, inplace=True)
    (13): Dropout2d(p=0.25, inplace=False)
    (14): BatchNorm2d(128, eps=0.8, momentum=0.1, affine=True,
track_running_stats=True)
  )
  (adv_layer): Linear(in_features=8192, out_features=1, bias=True)
)
```

[53]:
```
# Initialize BCELoss function
#criterion = nn.BCELoss()
criterion = nn.MSELoss()

# Create batch of latent vectors that we will use to visualize
#  the progression of the generator
fixed_noise = torch.randn(128, nz, 1, 1, device=device)
print(fixed_noise.shape)
# Establish convention for real and fake labels during training
real_label = 1
fake_label = 0


# Setup Adam optimizers for both G and D
optimizerD = optim.Adam(netD.parameters(), lr=lr, betas=(beta1, 0.999))
optimizerG = optim.Adam(netG.parameters(), lr=lr, betas=(beta1, 0.999))


# setup sgd
optimizerD = optim.SGD(netD.parameters(), lr=lr)
optimizerG = optim.SGD(netG.parameters(), lr=lr)
```

```
torch.Size([128, 100, 1, 1])
```

[54]:
```
from torch.autograd import Variable
```

[55]:
```python
Tensor = torch.cuda.FloatTensor
img_list = []
G_losses = []
D_losses = []
iters = 0
# ----------
#  Training
# ----------

for epoch in range(num_epochs):
    for i, (imgs, _) in enumerate(dataloader):

        # Adversarial ground truths
        valid = Variable(Tensor(imgs.shape[0], 1).fill_(1.0),␣
↪requires_grad=False)
        fake = Variable(Tensor(imgs.shape[0], 1).fill_(0.0),␣
↪requires_grad=False)

        # Configure input
        real_imgs = Variable(imgs.type(Tensor))

        # -----------------
        #  Train Generator
        # -----------------

        optimizerG.zero_grad()

        # Sample noise as generator input
        z = Variable(Tensor(np.random.normal(0, 1, (imgs.shape[0], 100))))

        # Generate a batch of images
        gen_imgs = netG(z)
        #criterion = nn.MSELoss()
        # Loss measures generator's ability to fool the discriminator
        g_loss = 0.5*criterion(netD(gen_imgs), valid)

        g_loss.backward()
        optimizerG.step()

        # ---------------------
        #  Train Discriminator
        # ---------------------

        optimizerD.zero_grad()

        # Measure discriminator's ability to classify real from generated␣
↪samples
```

```
        real_loss = criterion(netD(real_imgs), valid)
        fake_loss = criterion(netD(gen_imgs.detach()), fake)
        d_loss = 0.5 * (real_loss + fake_loss)

        d_loss.backward()
        optimizerD.step()

        # Output training stats
        if i % 50 == 0:
            print('[%d/%d][%d/%d]\tLoss_D: %.4f\tLoss_G: %.4f'
                  % (epoch, num_epochs, i, len(dataloader),
                     d_loss.item(), g_loss.item()))

        # Save Losses for plotting later
        G_losses.append(g_loss.item())
        D_losses.append(d_loss.item())
        batches_done = epoch * len(dataloader) + i
        # Check how the generator is doing by saving G's output on fixed_noise
        if (iters % 100 == 0) or ((epoch == 50-1) and (i == len(dataloader)-1)):
            #with torch.no_grad():
                #fake = netG(fixed_noise).detach().cpu()
            #img_list.append(vutils.make_grid(fake, padding=0, normalize=True))
            save_image(gen_imgs.data[:25], directoryName+"/%d.png" %␣
↪batches_done, nrow=5, normalize=True)
#              torch.save(netG.state_dict(), directoryName+"/
↪Batch%d_netGModel_iter%d" % (batches_done, iters))
        iters += 1
```

```
[0/150][0/270]  Loss_D: 0.5013  Loss_G: 0.5011
[0/150][50/270] Loss_D: 0.2509  Loss_G: 0.1341
[0/150][100/270]        Loss_D: 0.2504  Loss_G: 0.1254
[0/150][150/270]        Loss_D: 0.2504  Loss_G: 0.1254
[0/150][200/270]        Loss_D: 0.2503  Loss_G: 0.1254
[0/150][250/270]        Loss_D: 0.2504  Loss_G: 0.1256
[1/150][0/270]  Loss_D: 0.2502  Loss_G: 0.1255
[1/150][50/270] Loss_D: 0.2501  Loss_G: 0.1251
[1/150][100/270]        Loss_D: 0.2500  Loss_G: 0.1254
[1/150][150/270]        Loss_D: 0.2501  Loss_G: 0.1254
[1/150][200/270]        Loss_D: 0.2503  Loss_G: 0.1252
[1/150][250/270]        Loss_D: 0.2502  Loss_G: 0.1254
[2/150][0/270]  Loss_D: 0.2497  Loss_G: 0.1254
[2/150][50/270] Loss_D: 0.2497  Loss_G: 0.1254
[2/150][100/270]        Loss_D: 0.2496  Loss_G: 0.1256
[2/150][150/270]        Loss_D: 0.2495  Loss_G: 0.1253
[2/150][200/270]        Loss_D: 0.2493  Loss_G: 0.1255
[2/150][250/270]        Loss_D: 0.2497  Loss_G: 0.1258
[3/150][0/270]  Loss_D: 0.2496  Loss_G: 0.1258
```

```
[3/150][50/270] Loss_D: 0.2494  Loss_G: 0.1258
[3/150][100/270]        Loss_D: 0.2491  Loss_G: 0.1259
[3/150][150/270]        Loss_D: 0.2490  Loss_G: 0.1257
[3/150][200/270]        Loss_D: 0.2484  Loss_G: 0.1262
[3/150][250/270]        Loss_D: 0.2475  Loss_G: 0.1264
[4/150][0/270]  Loss_D: 0.2476  Loss_G: 0.1263
[4/150][50/270] Loss_D: 0.2452  Loss_G: 0.1269
[4/150][100/270]        Loss_D: 0.2436  Loss_G: 0.1281
[4/150][150/270]        Loss_D: 0.2411  Loss_G: 0.1297
[4/150][200/270]        Loss_D: 0.2294  Loss_G: 0.1340
[4/150][250/270]        Loss_D: 0.2100  Loss_G: 0.1439
[5/150][0/270]  Loss_D: 0.1997  Loss_G: 0.1532
[5/150][50/270] Loss_D: 0.1546  Loss_G: 0.1662
[5/150][100/270]        Loss_D: 0.1535  Loss_G: 0.1758
[5/150][150/270]        Loss_D: 0.2983  Loss_G: 0.1376
[5/150][200/270]        Loss_D: 0.1979  Loss_G: 0.2100
[5/150][250/270]        Loss_D: 0.2122  Loss_G: 0.1309
[6/150][0/270]  Loss_D: 0.2170  Loss_G: 0.1383
[6/150][50/270] Loss_D: 0.2349  Loss_G: 0.1426
[6/150][100/270]        Loss_D: 0.2359  Loss_G: 0.1297
[6/150][150/270]        Loss_D: 0.2666  Loss_G: 0.1379
[6/150][200/270]        Loss_D: 0.2297  Loss_G: 0.1309
[6/150][250/270]        Loss_D: 0.2364  Loss_G: 0.1342
[7/150][0/270]  Loss_D: 0.2529  Loss_G: 0.1168
[7/150][50/270] Loss_D: 0.2568  Loss_G: 0.1310
[7/150][100/270]        Loss_D: 0.2546  Loss_G: 0.1343
[7/150][150/270]        Loss_D: 0.2409  Loss_G: 0.1340
[7/150][200/270]        Loss_D: 0.2444  Loss_G: 0.1351
[7/150][250/270]        Loss_D: 0.2280  Loss_G: 0.1562
[8/150][0/270]  Loss_D: 0.2427  Loss_G: 0.1257
[8/150][50/270] Loss_D: 0.1921  Loss_G: 0.1470
[8/150][100/270]        Loss_D: 0.2140  Loss_G: 0.1538
[8/150][150/270]        Loss_D: 0.2186  Loss_G: 0.1437
[8/150][200/270]        Loss_D: 0.2243  Loss_G: 0.1281
[8/150][250/270]        Loss_D: 0.1430  Loss_G: 0.1648
[9/150][0/270]  Loss_D: 0.2331  Loss_G: 0.3297
[9/150][50/270] Loss_D: 0.1591  Loss_G: 0.1977
[9/150][100/270]        Loss_D: 0.1305  Loss_G: 0.3223
[9/150][150/270]        Loss_D: 0.0800  Loss_G: 0.1544
[9/150][200/270]        Loss_D: 0.0843  Loss_G: 0.2439
[9/150][250/270]        Loss_D: 0.0820  Loss_G: 0.2806
[10/150][0/270] Loss_D: 0.0976  Loss_G: 0.2329
[10/150][50/270]        Loss_D: 0.0606  Loss_G: 0.3304
[10/150][100/270]       Loss_D: 0.0541  Loss_G: 0.3033
[10/150][150/270]       Loss_D: 0.0601  Loss_G: 0.2037
[10/150][200/270]       Loss_D: 0.0887  Loss_G: 0.1864
[10/150][250/270]       Loss_D: 0.1124  Loss_G: 0.0906
[11/150][0/270] Loss_D: 0.1182  Loss_G: 0.1020
```

```
[11/150][50/270]        Loss_D: 0.0587  Loss_G: 0.5969
[11/150][100/270]       Loss_D: 0.0238  Loss_G: 0.3558
[11/150][150/270]       Loss_D: 0.0918  Loss_G: 0.2927
[11/150][200/270]       Loss_D: 0.2578  Loss_G: 0.2580
[11/150][250/270]       Loss_D: 0.1026  Loss_G: 0.5048
[12/150][0/270] Loss_D: 0.0937  Loss_G: 0.1285
[12/150][50/270]        Loss_D: 0.0349  Loss_G: 0.4113
[12/150][100/270]       Loss_D: 0.0298  Loss_G: 0.5922
[12/150][150/270]       Loss_D: 0.0564  Loss_G: 0.5050
[12/150][200/270]       Loss_D: 0.0802  Loss_G: 0.4580
[12/150][250/270]       Loss_D: 0.0303  Loss_G: 0.2974
[13/150][0/270] Loss_D: 0.0768  Loss_G: 0.2943
[13/150][50/270]        Loss_D: 0.0315  Loss_G: 0.4430
[13/150][100/270]       Loss_D: 0.0091  Loss_G: 0.3853
[13/150][150/270]       Loss_D: 0.1072  Loss_G: 0.3109
[13/150][200/270]       Loss_D: 0.0423  Loss_G: 0.3420
[13/150][250/270]       Loss_D: 0.1243  Loss_G: 0.4375
[14/150][0/270] Loss_D: 0.0644  Loss_G: 0.1839
[14/150][50/270]        Loss_D: 0.0236  Loss_G: 0.4797
[14/150][100/270]       Loss_D: 0.0463  Loss_G: 0.4371
[14/150][150/270]       Loss_D: 0.0250  Loss_G: 0.3432
[14/150][200/270]       Loss_D: 0.0163  Loss_G: 0.5010
[14/150][250/270]       Loss_D: 0.1228  Loss_G: 0.3898
[15/150][0/270] Loss_D: 0.0510  Loss_G: 0.4153
[15/150][50/270]        Loss_D: 0.0746  Loss_G: 0.5055
[15/150][100/270]       Loss_D: 0.0937  Loss_G: 0.2110
[15/150][150/270]       Loss_D: 0.0535  Loss_G: 0.2843
[15/150][200/270]       Loss_D: 0.2400  Loss_G: 0.2077
[15/150][250/270]       Loss_D: 0.2479  Loss_G: 0.1946
[16/150][0/270] Loss_D: 0.0720  Loss_G: 0.2561
[16/150][50/270]        Loss_D: 0.0703  Loss_G: 0.1271
[16/150][100/270]       Loss_D: 0.1553  Loss_G: 0.2627
[16/150][150/270]       Loss_D: 0.0532  Loss_G: 0.4243
[16/150][200/270]       Loss_D: 0.0309  Loss_G: 0.2641
[16/150][250/270]       Loss_D: 0.0941  Loss_G: 0.3441
[17/150][0/270] Loss_D: 0.1403  Loss_G: 0.2293
[17/150][50/270]        Loss_D: 0.1004  Loss_G: 0.3547
[17/150][100/270]       Loss_D: 0.1091  Loss_G: 0.1362
[17/150][150/270]       Loss_D: 0.0361  Loss_G: 0.4699
[17/150][200/270]       Loss_D: 0.1205  Loss_G: 0.1816
[17/150][250/270]       Loss_D: 0.1084  Loss_G: 0.3897
[18/150][0/270] Loss_D: 0.2097  Loss_G: 0.1703
[18/150][50/270]        Loss_D: 0.0337  Loss_G: 0.7114
[18/150][100/270]       Loss_D: 0.1873  Loss_G: 0.1702
[18/150][150/270]       Loss_D: 0.1581  Loss_G: 0.6967
[18/150][200/270]       Loss_D: 0.0724  Loss_G: 0.2828
[18/150][250/270]       Loss_D: 0.0804  Loss_G: 0.5060
[19/150][0/270] Loss_D: 0.1010  Loss_G: 0.2090
```

```
[19/150][50/270]        Loss_D: 0.1057  Loss_G: 0.2395
[19/150][100/270]       Loss_D: 0.1032  Loss_G: 0.2104
[19/150][150/270]       Loss_D: 0.0276  Loss_G: 0.5149
[19/150][200/270]       Loss_D: 0.0396  Loss_G: 0.4031
[19/150][250/270]       Loss_D: 0.0453  Loss_G: 0.2759
[20/150][0/270] Loss_D: 0.0432  Loss_G: 0.3717
[20/150][50/270]        Loss_D: 0.0937  Loss_G: 0.3865
[20/150][100/270]       Loss_D: 0.0163  Loss_G: 0.4472
[20/150][150/270]       Loss_D: 0.0189  Loss_G: 0.7561
[20/150][200/270]       Loss_D: 0.0498  Loss_G: 0.3049
[20/150][250/270]       Loss_D: 0.0864  Loss_G: 0.3983
[21/150][0/270] Loss_D: 0.0905  Loss_G: 0.1571
[21/150][50/270]        Loss_D: 0.0346  Loss_G: 0.4079
[21/150][100/270]       Loss_D: 0.0721  Loss_G: 0.2317
[21/150][150/270]       Loss_D: 0.0628  Loss_G: 0.2815
[21/150][200/270]       Loss_D: 0.0130  Loss_G: 0.6008
[21/150][250/270]       Loss_D: 0.0366  Loss_G: 0.2869
[22/150][0/270] Loss_D: 0.0492  Loss_G: 0.2745
[22/150][50/270]        Loss_D: 0.0494  Loss_G: 0.5154
[22/150][100/270]       Loss_D: 0.0237  Loss_G: 0.3479
[22/150][150/270]       Loss_D: 0.1762  Loss_G: 0.6183
[22/150][200/270]       Loss_D: 0.0329  Loss_G: 0.4863
[22/150][250/270]       Loss_D: 0.0459  Loss_G: 0.3903
[23/150][0/270] Loss_D: 0.0776  Loss_G: 0.2601
[23/150][50/270]        Loss_D: 0.1682  Loss_G: 0.1121
[23/150][100/270]       Loss_D: 0.0100  Loss_G: 0.3986
[23/150][150/270]       Loss_D: 0.0127  Loss_G: 0.4740
[23/150][200/270]       Loss_D: 0.0276  Loss_G: 0.4508
[23/150][250/270]       Loss_D: 0.0179  Loss_G: 0.4551
[24/150][0/270] Loss_D: 0.0649  Loss_G: 0.4034
[24/150][50/270]        Loss_D: 0.0143  Loss_G: 0.4256
[24/150][100/270]       Loss_D: 0.0300  Loss_G: 0.5862
[24/150][150/270]       Loss_D: 0.0222  Loss_G: 0.4883
[24/150][200/270]       Loss_D: 0.0697  Loss_G: 0.3453
[24/150][250/270]       Loss_D: 0.0414  Loss_G: 0.3666
[25/150][0/270] Loss_D: 0.1244  Loss_G: 0.1240
[25/150][50/270]        Loss_D: 0.0391  Loss_G: 0.3383
[25/150][100/270]       Loss_D: 0.0165  Loss_G: 0.6440
[25/150][150/270]       Loss_D: 0.0247  Loss_G: 0.4512
[25/150][200/270]       Loss_D: 0.1936  Loss_G: 0.2697
[25/150][250/270]       Loss_D: 0.0359  Loss_G: 0.3941
[26/150][0/270] Loss_D: 0.1119  Loss_G: 0.1469
[26/150][50/270]        Loss_D: 0.0330  Loss_G: 0.4155
[26/150][100/270]       Loss_D: 0.1072  Loss_G: 0.2044
[26/150][150/270]       Loss_D: 0.0834  Loss_G: 0.5420
[26/150][200/270]       Loss_D: 0.0281  Loss_G: 0.4199
[26/150][250/270]       Loss_D: 0.0216  Loss_G: 0.5538
[27/150][0/270] Loss_D: 0.0710  Loss_G: 0.1821
```

```
[27/150][50/270]        Loss_D: 0.0260  Loss_G: 0.4093
[27/150][100/270]       Loss_D: 0.1231  Loss_G: 0.4855
[27/150][150/270]       Loss_D: 0.0320  Loss_G: 0.2648
[27/150][200/270]       Loss_D: 0.0293  Loss_G: 0.2488
[27/150][250/270]       Loss_D: 0.1026  Loss_G: 0.3933
[28/150][0/270] Loss_D: 0.0506  Loss_G: 0.1198
[28/150][50/270]        Loss_D: 0.2495  Loss_G: 0.4780
[28/150][100/270]       Loss_D: 0.0363  Loss_G: 0.2872
[28/150][150/270]       Loss_D: 0.0588  Loss_G: 0.2784
[28/150][200/270]       Loss_D: 0.0354  Loss_G: 0.5300
[28/150][250/270]       Loss_D: 0.0566  Loss_G: 0.4471
[29/150][0/270] Loss_D: 0.0866  Loss_G: 0.2206
[29/150][50/270]        Loss_D: 0.0126  Loss_G: 0.4977
[29/150][100/270]       Loss_D: 0.0290  Loss_G: 0.5334
[29/150][150/270]       Loss_D: 0.0102  Loss_G: 0.5630
[29/150][200/270]       Loss_D: 0.0222  Loss_G: 0.3204
[29/150][250/270]       Loss_D: 0.0300  Loss_G: 0.4309
[30/150][0/270] Loss_D: 0.0137  Loss_G: 0.4741
[30/150][50/270]        Loss_D: 0.0178  Loss_G: 0.3822
[30/150][100/270]       Loss_D: 0.2060  Loss_G: 0.2485
[30/150][150/270]       Loss_D: 0.0847  Loss_G: 0.4932
[30/150][200/270]       Loss_D: 0.0570  Loss_G: 0.3040
[30/150][250/270]       Loss_D: 0.0531  Loss_G: 0.4557
[31/150][0/270] Loss_D: 0.0439  Loss_G: 0.2269
[31/150][50/270]        Loss_D: 0.0118  Loss_G: 0.6244
[31/150][100/270]       Loss_D: 0.0230  Loss_G: 0.4483
[31/150][150/270]       Loss_D: 0.0217  Loss_G: 0.3236
[31/150][200/270]       Loss_D: 0.0659  Loss_G: 0.3072
[31/150][250/270]       Loss_D: 0.0194  Loss_G: 0.4104
[32/150][0/270] Loss_D: 0.0539  Loss_G: 0.3275
[32/150][50/270]        Loss_D: 0.0322  Loss_G: 0.4755
[32/150][100/270]       Loss_D: 0.0058  Loss_G: 0.4655
[32/150][150/270]       Loss_D: 0.0296  Loss_G: 0.2891
[32/150][200/270]       Loss_D: 0.0891  Loss_G: 0.4335
[32/150][250/270]       Loss_D: 0.0640  Loss_G: 0.2287
[33/150][0/270] Loss_D: 0.0358  Loss_G: 0.2820
[33/150][50/270]        Loss_D: 0.0832  Loss_G: 0.4536
[33/150][100/270]       Loss_D: 0.0485  Loss_G: 0.2801
[33/150][150/270]       Loss_D: 0.0699  Loss_G: 0.3022
[33/150][200/270]       Loss_D: 0.0328  Loss_G: 0.3742
[33/150][250/270]       Loss_D: 0.0165  Loss_G: 0.4077
[34/150][0/270] Loss_D: 0.0488  Loss_G: 0.5114
[34/150][50/270]        Loss_D: 0.0313  Loss_G: 0.5409
[34/150][100/270]       Loss_D: 0.0697  Loss_G: 0.5483
[34/150][150/270]       Loss_D: 0.0358  Loss_G: 0.3786
[34/150][200/270]       Loss_D: 0.0979  Loss_G: 0.5488
[34/150][250/270]       Loss_D: 0.0297  Loss_G: 0.3586
[35/150][0/270] Loss_D: 0.0426  Loss_G: 0.3177
```

```
[35/150][50/270]        Loss_D: 0.0043  Loss_G: 0.5802
[35/150][100/270]       Loss_D: 0.1816  Loss_G: 0.5809
[35/150][150/270]       Loss_D: 0.0462  Loss_G: 0.3217
[35/150][200/270]       Loss_D: 0.1183  Loss_G: 0.3366
[35/150][250/270]       Loss_D: 0.0214  Loss_G: 0.5222
[36/150][0/270] Loss_D: 0.0582  Loss_G: 0.1663
[36/150][50/270]        Loss_D: 0.1293  Loss_G: 0.1966
[36/150][100/270]       Loss_D: 0.0357  Loss_G: 0.3986
[36/150][150/270]       Loss_D: 0.0235  Loss_G: 0.3861
[36/150][200/270]       Loss_D: 0.0660  Loss_G: 0.3518
[36/150][250/270]       Loss_D: 0.1433  Loss_G: 0.4900
[37/150][0/270] Loss_D: 0.0513  Loss_G: 0.3370
[37/150][50/270]        Loss_D: 0.0144  Loss_G: 0.4568
[37/150][100/270]       Loss_D: 0.0791  Loss_G: 0.4288
[37/150][150/270]       Loss_D: 0.0134  Loss_G: 0.2923
[37/150][200/270]       Loss_D: 0.0252  Loss_G: 0.3056
[37/150][250/270]       Loss_D: 0.0511  Loss_G: 0.3551
[38/150][0/270] Loss_D: 0.0630  Loss_G: 0.3390
[38/150][50/270]        Loss_D: 0.0348  Loss_G: 0.4087
[38/150][100/270]       Loss_D: 0.0063  Loss_G: 0.4701
[38/150][150/270]       Loss_D: 0.0122  Loss_G: 0.4265
[38/150][200/270]       Loss_D: 0.0094  Loss_G: 0.6178
[38/150][250/270]       Loss_D: 0.0093  Loss_G: 0.4627
[39/150][0/270] Loss_D: 0.0521  Loss_G: 0.2294
[39/150][50/270]        Loss_D: 0.0267  Loss_G: 0.4442
[39/150][100/270]       Loss_D: 0.0186  Loss_G: 0.3410
[39/150][150/270]       Loss_D: 0.0229  Loss_G: 0.4999
[39/150][200/270]       Loss_D: 0.0277  Loss_G: 0.3144
[39/150][250/270]       Loss_D: 0.0290  Loss_G: 0.3782
[40/150][0/270] Loss_D: 0.1370  Loss_G: 0.1334
[40/150][50/270]        Loss_D: 0.1072  Loss_G: 0.5380
[40/150][100/270]       Loss_D: 0.0182  Loss_G: 0.4895
[40/150][150/270]       Loss_D: 0.0129  Loss_G: 0.4749
[40/150][200/270]       Loss_D: 0.0195  Loss_G: 0.5260
[40/150][250/270]       Loss_D: 0.0142  Loss_G: 0.4189
[41/150][0/270] Loss_D: 0.0128  Loss_G: 0.4608
[41/150][50/270]        Loss_D: 0.0071  Loss_G: 0.4318
[41/150][100/270]       Loss_D: 0.0318  Loss_G: 0.2631
[41/150][150/270]       Loss_D: 0.0159  Loss_G: 0.4474
[41/150][200/270]       Loss_D: 0.0138  Loss_G: 0.3990
[41/150][250/270]       Loss_D: 0.0049  Loss_G: 0.4687
[42/150][0/270] Loss_D: 0.0219  Loss_G: 0.3362
[42/150][50/270]        Loss_D: 0.0264  Loss_G: 0.4829
[42/150][100/270]       Loss_D: 0.0309  Loss_G: 0.4908
[42/150][150/270]       Loss_D: 0.0095  Loss_G: 0.4890
[42/150][200/270]       Loss_D: 0.0037  Loss_G: 0.4438
[42/150][250/270]       Loss_D: 0.0189  Loss_G: 0.4518
[43/150][0/270] Loss_D: 0.0264  Loss_G: 0.2970
```

```
[43/150][50/270]        Loss_D: 0.0196  Loss_G: 0.5325
[43/150][100/270]       Loss_D: 0.0057  Loss_G: 0.3908
[43/150][150/270]       Loss_D: 0.0567  Loss_G: 0.3947
[43/150][200/270]       Loss_D: 0.0031  Loss_G: 0.4878
[43/150][250/270]       Loss_D: 0.0053  Loss_G: 0.4078
[44/150][0/270] Loss_D: 0.0092  Loss_G: 0.2270
[44/150][50/270]        Loss_D: 0.0342  Loss_G: 0.3464
[44/150][100/270]       Loss_D: 0.0153  Loss_G: 0.3009
[44/150][150/270]       Loss_D: 0.0688  Loss_G: 0.2435
[44/150][200/270]       Loss_D: 0.0238  Loss_G: 0.6156
[44/150][250/270]       Loss_D: 0.0665  Loss_G: 0.3336
[45/150][0/270] Loss_D: 0.0844  Loss_G: 0.1985
[45/150][50/270]        Loss_D: 0.3304  Loss_G: 0.2819
[45/150][100/270]       Loss_D: 0.0064  Loss_G: 0.5198
[45/150][150/270]       Loss_D: 0.0183  Loss_G: 0.5270
[45/150][200/270]       Loss_D: 0.0530  Loss_G: 0.3962
[45/150][250/270]       Loss_D: 0.0244  Loss_G: 0.4426
[46/150][0/270] Loss_D: 0.0560  Loss_G: 0.1872
[46/150][50/270]        Loss_D: 0.0767  Loss_G: 0.1868
[46/150][100/270]       Loss_D: 0.0126  Loss_G: 0.3904
[46/150][150/270]       Loss_D: 0.1032  Loss_G: 0.3713
[46/150][200/270]       Loss_D: 0.0141  Loss_G: 0.4327
[46/150][250/270]       Loss_D: 0.0108  Loss_G: 0.5952
[47/150][0/270] Loss_D: 0.0133  Loss_G: 0.3590
[47/150][50/270]        Loss_D: 0.0257  Loss_G: 0.4297
[47/150][100/270]       Loss_D: 0.0118  Loss_G: 0.4644
[47/150][150/270]       Loss_D: 0.0472  Loss_G: 0.4294
[47/150][200/270]       Loss_D: 0.1039  Loss_G: 0.4461
[47/150][250/270]       Loss_D: 0.0894  Loss_G: 0.4443
[48/150][0/270] Loss_D: 0.0605  Loss_G: 0.3062
[48/150][50/270]        Loss_D: 0.0496  Loss_G: 0.4699
[48/150][100/270]       Loss_D: 0.0193  Loss_G: 0.4635
[48/150][150/270]       Loss_D: 0.0134  Loss_G: 0.5167
[48/150][200/270]       Loss_D: 0.0198  Loss_G: 0.5313
[48/150][250/270]       Loss_D: 0.0065  Loss_G: 0.5148
[49/150][0/270] Loss_D: 0.0817  Loss_G: 0.1945
[49/150][50/270]        Loss_D: 0.0076  Loss_G: 0.5507
[49/150][100/270]       Loss_D: 0.0047  Loss_G: 0.3886
[49/150][150/270]       Loss_D: 0.0121  Loss_G: 0.3493
[49/150][200/270]       Loss_D: 0.0168  Loss_G: 0.4962
[49/150][250/270]       Loss_D: 0.0069  Loss_G: 0.4469
[50/150][0/270] Loss_D: 0.0510  Loss_G: 0.2710
[50/150][50/270]        Loss_D: 0.0397  Loss_G: 0.3936
[50/150][100/270]       Loss_D: 0.0092  Loss_G: 0.5106
[50/150][150/270]       Loss_D: 0.0675  Loss_G: 0.4816
[50/150][200/270]       Loss_D: 0.0081  Loss_G: 0.4501
[50/150][250/270]       Loss_D: 0.0187  Loss_G: 0.4599
[51/150][0/270] Loss_D: 0.0688  Loss_G: 0.2312
```

```
[51/150][50/270]        Loss_D: 0.0093  Loss_G: 0.5204
[51/150][100/270]       Loss_D: 0.0059  Loss_G: 0.5394
[51/150][150/270]       Loss_D: 0.0077  Loss_G: 0.5133
[51/150][200/270]       Loss_D: 0.0356  Loss_G: 0.5328
[51/150][250/270]       Loss_D: 0.0069  Loss_G: 0.7005
[52/150][0/270] Loss_D: 0.1210  Loss_G: 0.4911
[52/150][50/270]        Loss_D: 0.0335  Loss_G: 0.3275
[52/150][100/270]       Loss_D: 0.0056  Loss_G: 0.4631
[52/150][150/270]       Loss_D: 0.0107  Loss_G: 0.5448
[52/150][200/270]       Loss_D: 0.0298  Loss_G: 0.2517
[52/150][250/270]       Loss_D: 0.0246  Loss_G: 0.5062
[53/150][0/270] Loss_D: 0.0168  Loss_G: 0.5567
[53/150][50/270]        Loss_D: 0.0225  Loss_G: 0.4772
[53/150][100/270]       Loss_D: 0.0060  Loss_G: 0.4718
[53/150][150/270]       Loss_D: 0.0164  Loss_G: 0.4282
[53/150][200/270]       Loss_D: 0.0147  Loss_G: 0.5466
[53/150][250/270]       Loss_D: 0.0146  Loss_G: 0.4467
[54/150][0/270] Loss_D: 0.0249  Loss_G: 0.3301
[54/150][50/270]        Loss_D: 0.0402  Loss_G: 0.3164
[54/150][100/270]       Loss_D: 0.0199  Loss_G: 0.6232
[54/150][150/270]       Loss_D: 0.0284  Loss_G: 0.4752
[54/150][200/270]       Loss_D: 0.0268  Loss_G: 0.5030
[54/150][250/270]       Loss_D: 0.0625  Loss_G: 0.3035
[55/150][0/270] Loss_D: 0.0738  Loss_G: 0.2817
[55/150][50/270]        Loss_D: 0.0407  Loss_G: 0.5187
[55/150][100/270]       Loss_D: 0.0484  Loss_G: 0.1810
[55/150][150/270]       Loss_D: 0.0691  Loss_G: 0.5132
[55/150][200/270]       Loss_D: 0.0499  Loss_G: 0.4027
[55/150][250/270]       Loss_D: 0.0271  Loss_G: 0.3665
[56/150][0/270] Loss_D: 0.0790  Loss_G: 0.2263
[56/150][50/270]        Loss_D: 0.0108  Loss_G: 0.4420
[56/150][100/270]       Loss_D: 0.0131  Loss_G: 0.5244
[56/150][150/270]       Loss_D: 0.0432  Loss_G: 0.3634
[56/150][200/270]       Loss_D: 0.2091  Loss_G: 0.5403
[56/150][250/270]       Loss_D: 0.0110  Loss_G: 0.5578
[57/150][0/270] Loss_D: 0.0748  Loss_G: 0.2664
[57/150][50/270]        Loss_D: 0.0145  Loss_G: 0.3625
[57/150][100/270]       Loss_D: 0.0087  Loss_G: 0.4424
[57/150][150/270]       Loss_D: 0.0120  Loss_G: 0.4079
[57/150][200/270]       Loss_D: 0.0091  Loss_G: 0.5292
[57/150][250/270]       Loss_D: 0.0206  Loss_G: 0.7473
[58/150][0/270] Loss_D: 0.0290  Loss_G: 0.3385
[58/150][50/270]        Loss_D: 0.0191  Loss_G: 0.5477
[58/150][100/270]       Loss_D: 0.0434  Loss_G: 0.5154
[58/150][150/270]       Loss_D: 0.0104  Loss_G: 0.5008
[58/150][200/270]       Loss_D: 0.0321  Loss_G: 0.4652
[58/150][250/270]       Loss_D: 0.0082  Loss_G: 0.4796
[59/150][0/270] Loss_D: 0.0335  Loss_G: 0.4021
```

```
[59/150][50/270]        Loss_D: 0.0054  Loss_G: 0.4945
[59/150][100/270]       Loss_D: 0.0065  Loss_G: 0.4867
[59/150][150/270]       Loss_D: 0.0066  Loss_G: 0.3894
[59/150][200/270]       Loss_D: 0.0237  Loss_G: 0.3668
[59/150][250/270]       Loss_D: 0.0362  Loss_G: 0.3127
[60/150][0/270] Loss_D: 0.0587  Loss_G: 0.2855
[60/150][50/270]        Loss_D: 0.0071  Loss_G: 0.4682
[60/150][100/270]       Loss_D: 0.0247  Loss_G: 0.4833
[60/150][150/270]       Loss_D: 0.0096  Loss_G: 0.4069
[60/150][200/270]       Loss_D: 0.0035  Loss_G: 0.5399
[60/150][250/270]       Loss_D: 0.0073  Loss_G: 0.4231
[61/150][0/270] Loss_D: 0.0069  Loss_G: 0.5515
[61/150][50/270]        Loss_D: 0.0099  Loss_G: 0.4717
[61/150][100/270]       Loss_D: 0.0345  Loss_G: 0.6019
[61/150][150/270]       Loss_D: 0.0342  Loss_G: 0.5696
[61/150][200/270]       Loss_D: 0.0363  Loss_G: 0.3351
[61/150][250/270]       Loss_D: 0.0351  Loss_G: 0.4541
[62/150][0/270] Loss_D: 0.0712  Loss_G: 0.3323
[62/150][50/270]        Loss_D: 0.0285  Loss_G: 0.5199
[62/150][100/270]       Loss_D: 0.0053  Loss_G: 0.4661
[62/150][150/270]       Loss_D: 0.0132  Loss_G: 0.4402
[62/150][200/270]       Loss_D: 0.0220  Loss_G: 0.4906
[62/150][250/270]       Loss_D: 0.0102  Loss_G: 0.4690
[63/150][0/270] Loss_D: 0.0375  Loss_G: 0.4521
[63/150][50/270]        Loss_D: 0.0381  Loss_G: 0.5126
[63/150][100/270]       Loss_D: 0.0075  Loss_G: 0.5032
[63/150][150/270]       Loss_D: 0.0047  Loss_G: 0.4869
[63/150][200/270]       Loss_D: 0.0091  Loss_G: 0.4852
[63/150][250/270]       Loss_D: 0.0069  Loss_G: 0.4800
[64/150][0/270] Loss_D: 0.0536  Loss_G: 0.4795
[64/150][50/270]        Loss_D: 0.0070  Loss_G: 0.4882
[64/150][100/270]       Loss_D: 0.0081  Loss_G: 0.4821
[64/150][150/270]       Loss_D: 0.0111  Loss_G: 0.5071
[64/150][200/270]       Loss_D: 0.0240  Loss_G: 0.4760
[64/150][250/270]       Loss_D: 0.0045  Loss_G: 0.4397
[65/150][0/270] Loss_D: 0.1325  Loss_G: 0.1483
[65/150][50/270]        Loss_D: 0.0655  Loss_G: 0.4278
[65/150][100/270]       Loss_D: 0.0077  Loss_G: 0.4784
[65/150][150/270]       Loss_D: 0.0218  Loss_G: 0.2578
[65/150][200/270]       Loss_D: 0.0084  Loss_G: 0.5151
[65/150][250/270]       Loss_D: 0.0072  Loss_G: 0.4049
[66/150][0/270] Loss_D: 0.0169  Loss_G: 0.3854
[66/150][50/270]        Loss_D: 0.0133  Loss_G: 0.5601
[66/150][100/270]       Loss_D: 0.0580  Loss_G: 0.5597
[66/150][150/270]       Loss_D: 0.0089  Loss_G: 0.4021
[66/150][200/270]       Loss_D: 0.0207  Loss_G: 0.5885
[66/150][250/270]       Loss_D: 0.0487  Loss_G: 0.4546
[67/150][0/270] Loss_D: 0.0558  Loss_G: 0.3195
```

```
[67/150][50/270]        Loss_D: 0.0319  Loss_G: 0.4414
[67/150][100/270]       Loss_D: 0.0202  Loss_G: 0.2905
[67/150][150/270]       Loss_D: 0.0260  Loss_G: 0.4250
[67/150][200/270]       Loss_D: 0.0058  Loss_G: 0.5100
[67/150][250/270]       Loss_D: 0.0397  Loss_G: 0.4330
[68/150][0/270] Loss_D: 0.1165  Loss_G: 0.1636
[68/150][50/270]        Loss_D: 0.0184  Loss_G: 0.4184
[68/150][100/270]       Loss_D: 0.0215  Loss_G: 0.4998
[68/150][150/270]       Loss_D: 0.0110  Loss_G: 0.4663
[68/150][200/270]       Loss_D: 0.0171  Loss_G: 0.6085
[68/150][250/270]       Loss_D: 0.0181  Loss_G: 0.4688
[69/150][0/270] Loss_D: 0.0227  Loss_G: 0.4620
[69/150][50/270]        Loss_D: 0.0066  Loss_G: 0.5462
[69/150][100/270]       Loss_D: 0.0050  Loss_G: 0.5606
[69/150][150/270]       Loss_D: 0.0225  Loss_G: 0.5371
[69/150][200/270]       Loss_D: 0.0063  Loss_G: 0.5273
[69/150][250/270]       Loss_D: 0.0070  Loss_G: 0.4747
[70/150][0/270] Loss_D: 0.0162  Loss_G: 0.4640
[70/150][50/270]        Loss_D: 0.0047  Loss_G: 0.4551
[70/150][100/270]       Loss_D: 0.0025  Loss_G: 0.4556
[70/150][150/270]       Loss_D: 0.0489  Loss_G: 0.4508
[70/150][200/270]       Loss_D: 0.0146  Loss_G: 0.5087
[70/150][250/270]       Loss_D: 0.0120  Loss_G: 0.4817
[71/150][0/270] Loss_D: 0.0058  Loss_G: 0.4343
[71/150][50/270]        Loss_D: 0.0049  Loss_G: 0.4837
[71/150][100/270]       Loss_D: 0.0059  Loss_G: 0.5041
[71/150][150/270]       Loss_D: 0.0317  Loss_G: 0.4700
[71/150][200/270]       Loss_D: 0.0065  Loss_G: 0.5140
[71/150][250/270]       Loss_D: 0.0075  Loss_G: 0.4552
[72/150][0/270] Loss_D: 0.0876  Loss_G: 0.3324
[72/150][50/270]        Loss_D: 0.0174  Loss_G: 0.4964
[72/150][100/270]       Loss_D: 0.0098  Loss_G: 0.2918
[72/150][150/270]       Loss_D: 0.0178  Loss_G: 0.5229
[72/150][200/270]       Loss_D: 0.0124  Loss_G: 0.5450
[72/150][250/270]       Loss_D: 0.0140  Loss_G: 0.5408
[73/150][0/270] Loss_D: 0.0210  Loss_G: 0.4061
[73/150][50/270]        Loss_D: 0.0184  Loss_G: 0.4706
[73/150][100/270]       Loss_D: 0.0156  Loss_G: 0.4370
[73/150][150/270]       Loss_D: 0.0261  Loss_G: 0.5311
[73/150][200/270]       Loss_D: 0.0158  Loss_G: 0.2224
[73/150][250/270]       Loss_D: 0.0375  Loss_G: 0.3906
[74/150][0/270] Loss_D: 0.0319  Loss_G: 0.4139
[74/150][50/270]        Loss_D: 0.0399  Loss_G: 0.5279
[74/150][100/270]       Loss_D: 0.0254  Loss_G: 0.4279
[74/150][150/270]       Loss_D: 0.0144  Loss_G: 0.6080
[74/150][200/270]       Loss_D: 0.0330  Loss_G: 0.6257
[74/150][250/270]       Loss_D: 0.0418  Loss_G: 0.3156
[75/150][0/270] Loss_D: 0.0604  Loss_G: 0.3931
```

```
[75/150][50/270]        Loss_D: 0.0695  Loss_G: 0.3637
[75/150][100/270]       Loss_D: 0.0058  Loss_G: 0.4581
[75/150][150/270]       Loss_D: 0.0298  Loss_G: 0.4120
[75/150][200/270]       Loss_D: 0.0133  Loss_G: 0.6594
[75/150][250/270]       Loss_D: 0.0044  Loss_G: 0.4652
[76/150][0/270] Loss_D: 0.0302  Loss_G: 0.3341
[76/150][50/270]        Loss_D: 0.0664  Loss_G: 0.4025
[76/150][100/270]       Loss_D: 0.0090  Loss_G: 0.5778
[76/150][150/270]       Loss_D: 0.0072  Loss_G: 0.6290
[76/150][200/270]       Loss_D: 0.0264  Loss_G: 0.3352
[76/150][250/270]       Loss_D: 0.0154  Loss_G: 0.5789
[77/150][0/270] Loss_D: 0.1866  Loss_G: 0.1434
[77/150][50/270]        Loss_D: 0.0235  Loss_G: 0.6006
[77/150][100/270]       Loss_D: 0.0098  Loss_G: 0.4142
[77/150][150/270]       Loss_D: 0.0325  Loss_G: 0.6375
[77/150][200/270]       Loss_D: 0.0214  Loss_G: 0.5267
[77/150][250/270]       Loss_D: 0.0124  Loss_G: 0.4523
[78/150][0/270] Loss_D: 0.0479  Loss_G: 0.2994
[78/150][50/270]        Loss_D: 0.0108  Loss_G: 0.4323
[78/150][100/270]       Loss_D: 0.0264  Loss_G: 0.2811
[78/150][150/270]       Loss_D: 0.0155  Loss_G: 0.4118
[78/150][200/270]       Loss_D: 0.0187  Loss_G: 0.4752
[78/150][250/270]       Loss_D: 0.0050  Loss_G: 0.5006
[79/150][0/270] Loss_D: 0.0481  Loss_G: 0.2452
[79/150][50/270]        Loss_D: 0.0154  Loss_G: 0.4902
[79/150][100/270]       Loss_D: 0.0285  Loss_G: 0.5359
[79/150][150/270]       Loss_D: 0.0130  Loss_G: 0.4981
[79/150][200/270]       Loss_D: 0.0080  Loss_G: 0.4476
[79/150][250/270]       Loss_D: 0.0088  Loss_G: 0.4927
[80/150][0/270] Loss_D: 0.0617  Loss_G: 0.2081
[80/150][50/270]        Loss_D: 0.0104  Loss_G: 0.5144
[80/150][100/270]       Loss_D: 0.0531  Loss_G: 0.4964
[80/150][150/270]       Loss_D: 0.0051  Loss_G: 0.4629
[80/150][200/270]       Loss_D: 0.0088  Loss_G: 0.5190
[80/150][250/270]       Loss_D: 0.0079  Loss_G: 0.5097
[81/150][0/270] Loss_D: 0.0713  Loss_G: 0.2137
[81/150][50/270]        Loss_D: 0.0248  Loss_G: 0.5134
[81/150][100/270]       Loss_D: 0.0085  Loss_G: 0.4853
[81/150][150/270]       Loss_D: 0.0091  Loss_G: 0.5123
[81/150][200/270]       Loss_D: 0.0318  Loss_G: 0.4705
[81/150][250/270]       Loss_D: 0.0097  Loss_G: 0.4940
[82/150][0/270] Loss_D: 0.0780  Loss_G: 0.3800
[82/150][50/270]        Loss_D: 0.0077  Loss_G: 0.4387
[82/150][100/270]       Loss_D: 0.0052  Loss_G: 0.5348
[82/150][150/270]       Loss_D: 0.0069  Loss_G: 0.5363
[82/150][200/270]       Loss_D: 0.0154  Loss_G: 0.4908
[82/150][250/270]       Loss_D: 0.0070  Loss_G: 0.4925
[83/150][0/270] Loss_D: 0.0460  Loss_G: 0.2628
```

```
[83/150][50/270]        Loss_D: 0.0039  Loss_G: 0.4763
[83/150][100/270]       Loss_D: 0.0044  Loss_G: 0.4038
[83/150][150/270]       Loss_D: 0.0143  Loss_G: 0.4513
[83/150][200/270]       Loss_D: 0.0099  Loss_G: 0.5210
[83/150][250/270]       Loss_D: 0.0105  Loss_G: 0.2636
[84/150][0/270] Loss_D: 0.1150  Loss_G: 0.2620
[84/150][50/270]        Loss_D: 0.0205  Loss_G: 0.4623
[84/150][100/270]       Loss_D: 0.0208  Loss_G: 0.3651
[84/150][150/270]       Loss_D: 0.0337  Loss_G: 0.2676
[84/150][200/270]       Loss_D: 0.0130  Loss_G: 0.4556
[84/150][250/270]       Loss_D: 0.0066  Loss_G: 0.5117
[85/150][0/270] Loss_D: 0.0680  Loss_G: 0.3273
[85/150][50/270]        Loss_D: 0.0153  Loss_G: 0.4526
[85/150][100/270]       Loss_D: 0.0074  Loss_G: 0.4500
[85/150][150/270]       Loss_D: 0.0094  Loss_G: 0.4785
[85/150][200/270]       Loss_D: 0.0768  Loss_G: 0.2464
[85/150][250/270]       Loss_D: 0.1415  Loss_G: 0.1243
[86/150][0/270] Loss_D: 0.0227  Loss_G: 0.3508
[86/150][50/270]        Loss_D: 0.0275  Loss_G: 0.3780
[86/150][100/270]       Loss_D: 0.0133  Loss_G: 0.4048
[86/150][150/270]       Loss_D: 0.0118  Loss_G: 0.4746
[86/150][200/270]       Loss_D: 0.0121  Loss_G: 0.4170
[86/150][250/270]       Loss_D: 0.0069  Loss_G: 0.4842
[87/150][0/270] Loss_D: 0.0531  Loss_G: 0.2867
[87/150][50/270]        Loss_D: 0.0049  Loss_G: 0.4989
[87/150][100/270]       Loss_D: 0.0167  Loss_G: 0.5211
[87/150][150/270]       Loss_D: 0.0038  Loss_G: 0.4223
[87/150][200/270]       Loss_D: 0.0104  Loss_G: 0.4221
[87/150][250/270]       Loss_D: 0.0046  Loss_G: 0.5123
[88/150][0/270] Loss_D: 0.0269  Loss_G: 0.3452
[88/150][50/270]        Loss_D: 0.0056  Loss_G: 0.5012
[88/150][100/270]       Loss_D: 0.0060  Loss_G: 0.4890
[88/150][150/270]       Loss_D: 0.0075  Loss_G: 0.5415
[88/150][200/270]       Loss_D: 0.0117  Loss_G: 0.5531
[88/150][250/270]       Loss_D: 0.0093  Loss_G: 0.5011
[89/150][0/270] Loss_D: 0.0084  Loss_G: 0.4662
[89/150][50/270]        Loss_D: 0.0160  Loss_G: 0.4206
[89/150][100/270]       Loss_D: 0.0052  Loss_G: 0.4239
[89/150][150/270]       Loss_D: 0.0451  Loss_G: 0.3673
[89/150][200/270]       Loss_D: 0.0152  Loss_G: 0.3238
[89/150][250/270]       Loss_D: 0.0068  Loss_G: 0.5614
[90/150][0/270] Loss_D: 0.0182  Loss_G: 0.5378
[90/150][50/270]        Loss_D: 0.0181  Loss_G: 0.5048
[90/150][100/270]       Loss_D: 0.0280  Loss_G: 0.3800
[90/150][150/270]       Loss_D: 0.0171  Loss_G: 0.3393
[90/150][200/270]       Loss_D: 0.0077  Loss_G: 0.5469
[90/150][250/270]       Loss_D: 0.0032  Loss_G: 0.6446
[91/150][0/270] Loss_D: 0.0773  Loss_G: 0.2515
```

```
[91/150][50/270]        Loss_D: 0.0140  Loss_G: 0.6164
[91/150][100/270]       Loss_D: 0.0046  Loss_G: 0.3434
[91/150][150/270]       Loss_D: 0.0544  Loss_G: 0.5211
[91/150][200/270]       Loss_D: 0.0065  Loss_G: 0.3189
[91/150][250/270]       Loss_D: 0.0148  Loss_G: 0.4720
[92/150][0/270] Loss_D: 0.0724  Loss_G: 0.2307
[92/150][50/270]        Loss_D: 0.0155  Loss_G: 0.2564
[92/150][100/270]       Loss_D: 0.0188  Loss_G: 0.6237
[92/150][150/270]       Loss_D: 0.0031  Loss_G: 0.4741
[92/150][200/270]       Loss_D: 0.0125  Loss_G: 0.4366
[92/150][250/270]       Loss_D: 0.0241  Loss_G: 0.3160
[93/150][0/270] Loss_D: 0.0610  Loss_G: 0.2617
[93/150][50/270]        Loss_D: 0.0139  Loss_G: 0.4689
[93/150][100/270]       Loss_D: 0.0416  Loss_G: 0.5058
[93/150][150/270]       Loss_D: 0.0072  Loss_G: 0.4798
[93/150][200/270]       Loss_D: 0.0113  Loss_G: 0.4717
[93/150][250/270]       Loss_D: 0.0091  Loss_G: 0.4031
[94/150][0/270] Loss_D: 0.0678  Loss_G: 0.4501
[94/150][50/270]        Loss_D: 0.0206  Loss_G: 0.4244
[94/150][100/270]       Loss_D: 0.0101  Loss_G: 0.4306
[94/150][150/270]       Loss_D: 0.0101  Loss_G: 0.5863
[94/150][200/270]       Loss_D: 0.0058  Loss_G: 0.5265
[94/150][250/270]       Loss_D: 0.0073  Loss_G: 0.5202
[95/150][0/270] Loss_D: 0.0281  Loss_G: 0.3371
[95/150][50/270]        Loss_D: 0.0043  Loss_G: 0.4523
[95/150][100/270]       Loss_D: 0.0109  Loss_G: 0.5355
[95/150][150/270]       Loss_D: 0.0030  Loss_G: 0.4638
[95/150][200/270]       Loss_D: 0.0075  Loss_G: 0.5004
[95/150][250/270]       Loss_D: 0.0141  Loss_G: 0.4719
[96/150][0/270] Loss_D: 0.1013  Loss_G: 0.1802
[96/150][50/270]        Loss_D: 0.0035  Loss_G: 0.5073
[96/150][100/270]       Loss_D: 0.0045  Loss_G: 0.4392
[96/150][150/270]       Loss_D: 0.0231  Loss_G: 0.4752
[96/150][200/270]       Loss_D: 0.0052  Loss_G: 0.4545
[96/150][250/270]       Loss_D: 0.0037  Loss_G: 0.4851
[97/150][0/270] Loss_D: 0.0105  Loss_G: 0.3750
[97/150][50/270]        Loss_D: 0.0168  Loss_G: 0.3053
[97/150][100/270]       Loss_D: 0.0205  Loss_G: 0.3050
[97/150][150/270]       Loss_D: 0.0607  Loss_G: 0.7195
[97/150][200/270]       Loss_D: 0.1240  Loss_G: 0.4875
[97/150][250/270]       Loss_D: 0.0549  Loss_G: 0.2390
[98/150][0/270] Loss_D: 0.0168  Loss_G: 0.4109
[98/150][50/270]        Loss_D: 0.0376  Loss_G: 0.3488
[98/150][100/270]       Loss_D: 0.0649  Loss_G: 0.5372
[98/150][150/270]       Loss_D: 0.0085  Loss_G: 0.3318
[98/150][200/270]       Loss_D: 0.0246  Loss_G: 0.4787
[98/150][250/270]       Loss_D: 0.0050  Loss_G: 0.5106
[99/150][0/270] Loss_D: 0.0087  Loss_G: 0.5476
```

```
[99/150][50/270]        Loss_D: 0.0701  Loss_G: 0.3251
[99/150][100/270]       Loss_D: 0.0075  Loss_G: 0.4769
[99/150][150/270]       Loss_D: 0.0056  Loss_G: 0.5055
[99/150][200/270]       Loss_D: 0.0047  Loss_G: 0.5542
[99/150][250/270]       Loss_D: 0.0076  Loss_G: 0.5574
[100/150][0/270]        Loss_D: 0.0171  Loss_G: 0.4233
[100/150][50/270]       Loss_D: 0.0082  Loss_G: 0.5217
[100/150][100/270]      Loss_D: 0.0060  Loss_G: 0.4811
[100/150][150/270]      Loss_D: 0.0119  Loss_G: 0.4857
[100/150][200/270]      Loss_D: 0.0201  Loss_G: 0.3282
[100/150][250/270]      Loss_D: 0.0338  Loss_G: 0.4909
[101/150][0/270]        Loss_D: 0.0060  Loss_G: 0.4786
[101/150][50/270]       Loss_D: 0.0116  Loss_G: 0.5350
[101/150][100/270]      Loss_D: 0.0051  Loss_G: 0.5373
[101/150][150/270]      Loss_D: 0.0057  Loss_G: 0.4888
[101/150][200/270]      Loss_D: 0.0054  Loss_G: 0.4934
[101/150][250/270]      Loss_D: 0.0069  Loss_G: 0.5293
[102/150][0/270]        Loss_D: 0.0035  Loss_G: 0.4690
[102/150][50/270]       Loss_D: 0.0045  Loss_G: 0.4826
[102/150][100/270]      Loss_D: 0.0096  Loss_G: 0.5239
[102/150][150/270]      Loss_D: 0.0037  Loss_G: 0.5067
[102/150][200/270]      Loss_D: 0.0107  Loss_G: 0.4043
[102/150][250/270]      Loss_D: 0.0277  Loss_G: 0.7580
[103/150][0/270]        Loss_D: 0.0101  Loss_G: 0.3973
[103/150][50/270]       Loss_D: 0.0098  Loss_G: 0.4147
[103/150][100/270]      Loss_D: 0.0332  Loss_G: 0.5366
[103/150][150/270]      Loss_D: 0.0159  Loss_G: 0.5218
[103/150][200/270]      Loss_D: 0.0072  Loss_G: 0.4658
[103/150][250/270]      Loss_D: 0.0118  Loss_G: 0.5794
[104/150][0/270]        Loss_D: 0.0431  Loss_G: 0.3119
[104/150][50/270]       Loss_D: 0.0086  Loss_G: 0.5103
[104/150][100/270]      Loss_D: 0.0416  Loss_G: 0.4828
[104/150][150/270]      Loss_D: 0.0038  Loss_G: 0.5180
[104/150][200/270]      Loss_D: 0.0086  Loss_G: 0.5071
[104/150][250/270]      Loss_D: 0.0074  Loss_G: 0.4945
[105/150][0/270]        Loss_D: 0.0166  Loss_G: 0.3299
[105/150][50/270]       Loss_D: 0.0191  Loss_G: 0.4549
[105/150][100/270]      Loss_D: 0.0038  Loss_G: 0.4937
[105/150][150/270]      Loss_D: 0.0025  Loss_G: 0.4922
[105/150][200/270]      Loss_D: 0.0261  Loss_G: 0.3275
[105/150][250/270]      Loss_D: 0.0031  Loss_G: 0.4806
[106/150][0/270]        Loss_D: 0.0967  Loss_G: 0.2248
[106/150][50/270]       Loss_D: 0.0102  Loss_G: 0.4710
[106/150][100/270]      Loss_D: 0.0285  Loss_G: 0.4275
[106/150][150/270]      Loss_D: 0.0127  Loss_G: 0.5959
[106/150][200/270]      Loss_D: 0.0163  Loss_G: 0.6798
[106/150][250/270]      Loss_D: 0.0039  Loss_G: 0.5078
[107/150][0/270]        Loss_D: 0.0801  Loss_G: 0.4888
```

```
[107/150][50/270]     Loss_D: 0.0569  Loss_G: 0.3655
[107/150][100/270]    Loss_D: 0.0171  Loss_G: 0.6318
[107/150][150/270]    Loss_D: 0.0141  Loss_G: 0.3311
[107/150][200/270]    Loss_D: 0.0303  Loss_G: 0.5036
[107/150][250/270]    Loss_D: 0.0125  Loss_G: 0.7148
[108/150][0/270]      Loss_D: 0.0420  Loss_G: 0.3608
[108/150][50/270]     Loss_D: 0.0101  Loss_G: 0.5264
[108/150][100/270]    Loss_D: 0.0075  Loss_G: 0.4340
[108/150][150/270]    Loss_D: 0.0343  Loss_G: 0.5395
[108/150][200/270]    Loss_D: 0.0050  Loss_G: 0.4761
[108/150][250/270]    Loss_D: 0.0367  Loss_G: 0.3804
[109/150][0/270]      Loss_D: 0.0127  Loss_G: 0.5146
[109/150][50/270]     Loss_D: 0.0106  Loss_G: 0.4552
[109/150][100/270]    Loss_D: 0.0082  Loss_G: 0.4957
[109/150][150/270]    Loss_D: 0.0059  Loss_G: 0.5217
[109/150][200/270]    Loss_D: 0.0174  Loss_G: 0.4417
[109/150][250/270]    Loss_D: 0.0065  Loss_G: 0.4491
[110/150][0/270]      Loss_D: 0.0709  Loss_G: 0.2437
[110/150][50/270]     Loss_D: 0.0314  Loss_G: 0.4965
[110/150][100/270]    Loss_D: 0.0120  Loss_G: 0.5107
[110/150][150/270]    Loss_D: 0.0044  Loss_G: 0.5446
[110/150][200/270]    Loss_D: 0.0150  Loss_G: 0.5272
[110/150][250/270]    Loss_D: 0.0167  Loss_G: 0.4628
[111/150][0/270]      Loss_D: 0.0317  Loss_G: 0.4957
[111/150][50/270]     Loss_D: 0.0047  Loss_G: 0.5165
[111/150][100/270]    Loss_D: 0.0032  Loss_G: 0.5254
[111/150][150/270]    Loss_D: 0.0060  Loss_G: 0.5021
[111/150][200/270]    Loss_D: 0.0241  Loss_G: 0.4493
[111/150][250/270]    Loss_D: 0.0157  Loss_G: 0.3909
[112/150][0/270]      Loss_D: 0.0397  Loss_G: 0.3051
[112/150][50/270]     Loss_D: 0.0229  Loss_G: 0.6801
[112/150][100/270]    Loss_D: 0.0176  Loss_G: 0.4058
[112/150][150/270]    Loss_D: 0.0043  Loss_G: 0.4915
[112/150][200/270]    Loss_D: 0.1124  Loss_G: 0.3315
[112/150][250/270]    Loss_D: 0.0182  Loss_G: 0.5904
[113/150][0/270]      Loss_D: 0.0217  Loss_G: 0.3323
[113/150][50/270]     Loss_D: 0.0058  Loss_G: 0.5211
[113/150][100/270]    Loss_D: 0.0155  Loss_G: 0.4093
[113/150][150/270]    Loss_D: 0.0174  Loss_G: 0.4182
[113/150][200/270]    Loss_D: 0.0217  Loss_G: 0.5031
[113/150][250/270]    Loss_D: 0.0209  Loss_G: 0.4114
[114/150][0/270]      Loss_D: 0.0315  Loss_G: 0.4053
[114/150][50/270]     Loss_D: 0.0355  Loss_G: 0.5245
[114/150][100/270]    Loss_D: 0.0117  Loss_G: 0.5848
[114/150][150/270]    Loss_D: 0.0202  Loss_G: 0.3658
[114/150][200/270]    Loss_D: 0.0542  Loss_G: 0.5331
[114/150][250/270]    Loss_D: 0.0055  Loss_G: 0.5476
[115/150][0/270]      Loss_D: 0.0099  Loss_G: 0.4412
```

```
[115/150][50/270]       Loss_D: 0.0095  Loss_G: 0.5292
[115/150][100/270]      Loss_D: 0.0083  Loss_G: 0.4649
[115/150][150/270]      Loss_D: 0.0092  Loss_G: 0.5029
[115/150][200/270]      Loss_D: 0.0247  Loss_G: 0.4755
[115/150][250/270]      Loss_D: 0.0177  Loss_G: 0.4718
[116/150][0/270]        Loss_D: 0.0127  Loss_G: 0.4344
[116/150][50/270]       Loss_D: 0.0151  Loss_G: 0.5165
[116/150][100/270]      Loss_D: 0.0129  Loss_G: 0.4805
[116/150][150/270]      Loss_D: 0.0051  Loss_G: 0.5455
[116/150][200/270]      Loss_D: 0.0030  Loss_G: 0.5294
[116/150][250/270]      Loss_D: 0.0082  Loss_G: 0.4963
[117/150][0/270]        Loss_D: 0.0151  Loss_G: 0.3856
[117/150][50/270]       Loss_D: 0.0026  Loss_G: 0.4845
[117/150][100/270]      Loss_D: 0.0145  Loss_G: 0.4741
[117/150][150/270]      Loss_D: 0.0078  Loss_G: 0.5220
[117/150][200/270]      Loss_D: 0.0050  Loss_G: 0.4800
[117/150][250/270]      Loss_D: 0.0040  Loss_G: 0.4870
[118/150][0/270]        Loss_D: 0.0186  Loss_G: 0.3786
[118/150][50/270]       Loss_D: 0.0036  Loss_G: 0.4575
[118/150][100/270]      Loss_D: 0.0169  Loss_G: 0.4877
[118/150][150/270]      Loss_D: 0.0312  Loss_G: 0.4780
[118/150][200/270]      Loss_D: 0.0078  Loss_G: 0.6173
[118/150][250/270]      Loss_D: 0.0033  Loss_G: 0.4680
[119/150][0/270]        Loss_D: 0.0385  Loss_G: 0.1927
[119/150][50/270]       Loss_D: 0.0080  Loss_G: 0.4739
[119/150][100/270]      Loss_D: 0.0166  Loss_G: 0.4972
[119/150][150/270]      Loss_D: 0.0189  Loss_G: 0.4645
[119/150][200/270]      Loss_D: 0.0128  Loss_G: 0.4900
[119/150][250/270]      Loss_D: 0.0037  Loss_G: 0.5124
[120/150][0/270]        Loss_D: 0.0226  Loss_G: 0.3099
[120/150][50/270]       Loss_D: 0.0119  Loss_G: 0.2683
[120/150][100/270]      Loss_D: 0.0085  Loss_G: 0.5222
[120/150][150/270]      Loss_D: 0.0109  Loss_G: 0.3619
[120/150][200/270]      Loss_D: 0.0132  Loss_G: 0.2985
[120/150][250/270]      Loss_D: 0.0105  Loss_G: 0.2054
[121/150][0/270]        Loss_D: 0.0223  Loss_G: 0.3784
[121/150][50/270]       Loss_D: 0.0151  Loss_G: 0.5110
[121/150][100/270]      Loss_D: 0.0146  Loss_G: 0.4697
[121/150][150/270]      Loss_D: 0.0365  Loss_G: 0.3682
[121/150][200/270]      Loss_D: 0.0047  Loss_G: 0.2681
[121/150][250/270]      Loss_D: 0.0058  Loss_G: 0.5501
[122/150][0/270]        Loss_D: 0.0279  Loss_G: 0.3407
[122/150][50/270]       Loss_D: 0.0101  Loss_G: 0.4195
[122/150][100/270]      Loss_D: 0.0290  Loss_G: 0.3808
[122/150][150/270]      Loss_D: 0.0094  Loss_G: 0.3702
[122/150][200/270]      Loss_D: 0.0075  Loss_G: 0.5415
[122/150][250/270]      Loss_D: 0.0062  Loss_G: 0.4896
[123/150][0/270]        Loss_D: 0.0326  Loss_G: 0.4792
```

```
[123/150][50/270]     Loss_D: 0.0130  Loss_G: 0.4819
[123/150][100/270]    Loss_D: 0.0379  Loss_G: 0.3723
[123/150][150/270]    Loss_D: 0.0045  Loss_G: 0.5080
[123/150][200/270]    Loss_D: 0.0052  Loss_G: 0.4186
[123/150][250/270]    Loss_D: 0.0057  Loss_G: 0.5315
[124/150][0/270]      Loss_D: 0.0562  Loss_G: 0.2483
[124/150][50/270]     Loss_D: 0.0056  Loss_G: 0.4943
[124/150][100/270]    Loss_D: 0.0070  Loss_G: 0.4714
[124/150][150/270]    Loss_D: 0.0052  Loss_G: 0.4866
[124/150][200/270]    Loss_D: 0.0055  Loss_G: 0.5114
[124/150][250/270]    Loss_D: 0.0063  Loss_G: 0.4674
[125/150][0/270]      Loss_D: 0.1555  Loss_G: 0.1167
[125/150][50/270]     Loss_D: 0.0075  Loss_G: 0.4971
[125/150][100/270]    Loss_D: 0.0065  Loss_G: 0.4826
[125/150][150/270]    Loss_D: 0.0066  Loss_G: 0.5301
[125/150][200/270]    Loss_D: 0.0055  Loss_G: 0.4933
[125/150][250/270]    Loss_D: 0.0047  Loss_G: 0.4970
[126/150][0/270]      Loss_D: 0.0503  Loss_G: 0.1940
[126/150][50/270]     Loss_D: 0.0049  Loss_G: 0.5099
[126/150][100/270]    Loss_D: 0.0152  Loss_G: 0.5001
[126/150][150/270]    Loss_D: 0.0188  Loss_G: 0.5105
[126/150][200/270]    Loss_D: 0.0042  Loss_G: 0.4990
[126/150][250/270]    Loss_D: 0.0051  Loss_G: 0.4310
[127/150][0/270]      Loss_D: 0.0373  Loss_G: 0.2654
[127/150][50/270]     Loss_D: 0.0034  Loss_G: 0.4986
[127/150][100/270]    Loss_D: 0.0087  Loss_G: 0.4951
[127/150][150/270]    Loss_D: 0.0038  Loss_G: 0.4813
[127/150][200/270]    Loss_D: 0.0040  Loss_G: 0.5099
[127/150][250/270]    Loss_D: 0.0034  Loss_G: 0.4829
[128/150][0/270]      Loss_D: 0.0154  Loss_G: 0.4075
[128/150][50/270]     Loss_D: 0.0029  Loss_G: 0.5461
[128/150][100/270]    Loss_D: 0.0043  Loss_G: 0.4670
[128/150][150/270]    Loss_D: 0.0273  Loss_G: 0.5649
[128/150][200/270]    Loss_D: 0.0071  Loss_G: 0.4769
[128/150][250/270]    Loss_D: 0.0070  Loss_G: 0.5685
[129/150][0/270]      Loss_D: 0.0328  Loss_G: 0.5260
[129/150][50/270]     Loss_D: 0.0096  Loss_G: 0.4634
[129/150][100/270]    Loss_D: 0.0833  Loss_G: 0.5814
[129/150][150/270]    Loss_D: 0.0136  Loss_G: 0.6083
[129/150][200/270]    Loss_D: 0.0097  Loss_G: 0.5515
[129/150][250/270]    Loss_D: 0.0480  Loss_G: 0.2752
[130/150][0/270]      Loss_D: 0.0177  Loss_G: 0.6517
[130/150][50/270]     Loss_D: 0.0236  Loss_G: 0.4050
[130/150][100/270]    Loss_D: 0.0340  Loss_G: 0.2239
[130/150][150/270]    Loss_D: 0.0174  Loss_G: 0.5632
[130/150][200/270]    Loss_D: 0.0523  Loss_G: 0.5281
[130/150][250/270]    Loss_D: 0.0075  Loss_G: 0.5560
[131/150][0/270]      Loss_D: 0.0226  Loss_G: 0.6426
```

```
[131/150][50/270]       Loss_D: 0.1340  Loss_G: 0.4004
[131/150][100/270]      Loss_D: 0.0156  Loss_G: 0.5905
[131/150][150/270]      Loss_D: 0.0038  Loss_G: 0.4594
[131/150][200/270]      Loss_D: 0.0032  Loss_G: 0.4591
[131/150][250/270]      Loss_D: 0.0173  Loss_G: 0.5317
[132/150][0/270]        Loss_D: 0.0096  Loss_G: 0.4619
[132/150][50/270]       Loss_D: 0.0075  Loss_G: 0.4720
[132/150][100/270]      Loss_D: 0.0061  Loss_G: 0.5058
[132/150][150/270]      Loss_D: 0.0041  Loss_G: 0.4917
[132/150][200/270]      Loss_D: 0.0179  Loss_G: 0.5221
[132/150][250/270]      Loss_D: 0.0033  Loss_G: 0.4606
[133/150][0/270]        Loss_D: 0.0398  Loss_G: 0.2960
[133/150][50/270]       Loss_D: 0.0325  Loss_G: 0.4487
[133/150][100/270]      Loss_D: 0.0055  Loss_G: 0.4467
[133/150][150/270]      Loss_D: 0.0036  Loss_G: 0.4321
[133/150][200/270]      Loss_D: 0.0033  Loss_G: 0.4663
[133/150][250/270]      Loss_D: 0.0041  Loss_G: 0.5373
[134/150][0/270]        Loss_D: 0.0053  Loss_G: 0.4311
[134/150][50/270]       Loss_D: 0.0061  Loss_G: 0.4933
[134/150][100/270]      Loss_D: 0.0082  Loss_G: 0.5074
[134/150][150/270]      Loss_D: 0.0051  Loss_G: 0.4429
[134/150][200/270]      Loss_D: 0.0032  Loss_G: 0.4895
[134/150][250/270]      Loss_D: 0.0058  Loss_G: 0.4109
[135/150][0/270]        Loss_D: 0.0414  Loss_G: 0.2011
[135/150][50/270]       Loss_D: 0.0051  Loss_G: 0.4513
[135/150][100/270]      Loss_D: 0.0062  Loss_G: 0.4937
[135/150][150/270]      Loss_D: 0.0072  Loss_G: 0.4731
[135/150][200/270]      Loss_D: 0.0084  Loss_G: 0.4696
[135/150][250/270]      Loss_D: 0.0053  Loss_G: 0.4928
[136/150][0/270]        Loss_D: 0.0399  Loss_G: 0.3134
[136/150][50/270]       Loss_D: 0.0118  Loss_G: 0.4067
[136/150][100/270]      Loss_D: 0.0036  Loss_G: 0.5317
[136/150][150/270]      Loss_D: 0.0597  Loss_G: 0.3429
[136/150][200/270]      Loss_D: 0.0139  Loss_G: 0.5481
[136/150][250/270]      Loss_D: 0.0088  Loss_G: 0.4658
[137/150][0/270]        Loss_D: 0.0422  Loss_G: 0.2473
[137/150][50/270]       Loss_D: 0.0239  Loss_G: 0.4410
[137/150][100/270]      Loss_D: 0.0148  Loss_G: 0.5360
[137/150][150/270]      Loss_D: 0.0202  Loss_G: 0.4950
[137/150][200/270]      Loss_D: 0.0075  Loss_G: 0.4297
[137/150][250/270]      Loss_D: 0.0078  Loss_G: 0.4482
[138/150][0/270]        Loss_D: 0.0141  Loss_G: 0.4243
[138/150][50/270]       Loss_D: 0.0301  Loss_G: 0.2808
[138/150][100/270]      Loss_D: 0.0390  Loss_G: 0.5157
[138/150][150/270]      Loss_D: 0.0063  Loss_G: 0.5329
[138/150][200/270]      Loss_D: 0.0804  Loss_G: 0.3848
[138/150][250/270]      Loss_D: 0.0728  Loss_G: 0.3175
[139/150][0/270]        Loss_D: 0.0541  Loss_G: 0.7443
```

```
[139/150][50/270]       Loss_D: 0.0525  Loss_G: 0.5626
[139/150][100/270]      Loss_D: 0.0105  Loss_G: 0.4377
[139/150][150/270]      Loss_D: 0.0610  Loss_G: 0.3446
[139/150][200/270]      Loss_D: 0.0323  Loss_G: 0.4871
[139/150][250/270]      Loss_D: 0.0106  Loss_G: 0.4145
[140/150][0/270]        Loss_D: 0.0499  Loss_G: 0.3908
[140/150][50/270]       Loss_D: 0.0116  Loss_G: 0.4847
[140/150][100/270]      Loss_D: 0.0057  Loss_G: 0.4735
[140/150][150/270]      Loss_D: 0.0120  Loss_G: 0.4739
[140/150][200/270]      Loss_D: 0.0076  Loss_G: 0.5040
[140/150][250/270]      Loss_D: 0.0083  Loss_G: 0.5137
[141/150][0/270]        Loss_D: 0.0074  Loss_G: 0.4878
[141/150][50/270]       Loss_D: 0.0130  Loss_G: 0.4703
[141/150][100/270]      Loss_D: 0.0069  Loss_G: 0.4688
[141/150][150/270]      Loss_D: 0.0098  Loss_G: 0.4884
[141/150][200/270]      Loss_D: 0.0033  Loss_G: 0.5135
[141/150][250/270]      Loss_D: 0.0135  Loss_G: 0.5043
[142/150][0/270]        Loss_D: 0.0473  Loss_G: 0.2352
[142/150][50/270]       Loss_D: 0.0091  Loss_G: 0.5176
[142/150][100/270]      Loss_D: 0.0070  Loss_G: 0.5248
[142/150][150/270]      Loss_D: 0.0049  Loss_G: 0.4825
[142/150][200/270]      Loss_D: 0.0048  Loss_G: 0.4902
[142/150][250/270]      Loss_D: 0.0042  Loss_G: 0.4828
[143/150][0/270]        Loss_D: 0.0057  Loss_G: 0.3965
[143/150][50/270]       Loss_D: 0.0349  Loss_G: 0.3694
[143/150][100/270]      Loss_D: 0.0201  Loss_G: 0.4937
[143/150][150/270]      Loss_D: 0.0087  Loss_G: 0.3670
[143/150][200/270]      Loss_D: 0.0257  Loss_G: 0.6851
[143/150][250/270]      Loss_D: 0.0061  Loss_G: 0.4491
[144/150][0/270]        Loss_D: 0.0351  Loss_G: 0.3413
[144/150][50/270]       Loss_D: 0.0150  Loss_G: 0.3907
[144/150][100/270]      Loss_D: 0.0082  Loss_G: 0.5352
[144/150][150/270]      Loss_D: 0.0144  Loss_G: 0.6299
[144/150][200/270]      Loss_D: 0.0438  Loss_G: 0.6581
[144/150][250/270]      Loss_D: 0.0129  Loss_G: 0.6497
[145/150][0/270]        Loss_D: 0.0824  Loss_G: 0.2588
[145/150][50/270]       Loss_D: 0.0123  Loss_G: 0.4071
[145/150][100/270]      Loss_D: 0.0230  Loss_G: 0.4067
[145/150][150/270]      Loss_D: 0.0177  Loss_G: 0.3412
[145/150][200/270]      Loss_D: 0.0366  Loss_G: 0.4629
[145/150][250/270]      Loss_D: 0.0058  Loss_G: 0.4952
[146/150][0/270]        Loss_D: 0.0379  Loss_G: 0.7884
[146/150][50/270]       Loss_D: 0.0075  Loss_G: 0.5306
[146/150][100/270]      Loss_D: 0.0119  Loss_G: 0.3788
[146/150][150/270]      Loss_D: 0.0177  Loss_G: 0.4913
[146/150][200/270]      Loss_D: 0.0120  Loss_G: 0.6450
[146/150][250/270]      Loss_D: 0.0122  Loss_G: 0.4630
[147/150][0/270]        Loss_D: 0.0326  Loss_G: 0.3468
```

```
[147/150][50/270]       Loss_D: 0.0134  Loss_G: 0.4815
[147/150][100/270]      Loss_D: 0.0200  Loss_G: 0.4977
[147/150][150/270]      Loss_D: 0.0061  Loss_G: 0.4964
[147/150][200/270]      Loss_D: 0.0048  Loss_G: 0.4944
[147/150][250/270]      Loss_D: 0.0090  Loss_G: 0.5459
[148/150][0/270]        Loss_D: 0.0170  Loss_G: 0.4013
[148/150][50/270]       Loss_D: 0.0174  Loss_G: 0.4778
[148/150][100/270]      Loss_D: 0.0065  Loss_G: 0.4611
[148/150][150/270]      Loss_D: 0.0066  Loss_G: 0.5101
[148/150][200/270]      Loss_D: 0.0247  Loss_G: 0.4391
[148/150][250/270]      Loss_D: 0.0064  Loss_G: 0.5305
[149/150][0/270]        Loss_D: 0.0567  Loss_G: 0.2763
[149/150][50/270]       Loss_D: 0.0115  Loss_G: 0.5564
[149/150][100/270]      Loss_D: 0.0565  Loss_G: 0.4562
[149/150][150/270]      Loss_D: 0.0061  Loss_G: 0.4817
[149/150][200/270]      Loss_D: 0.0048  Loss_G: 0.4474
[149/150][250/270]      Loss_D: 0.0164  Loss_G: 0.4798
```

[36]:
```
print("/Batch%d_netGModel_iter%d" % (batches_done, iters))
```

```
/Batch269_netGModel_iter270
```

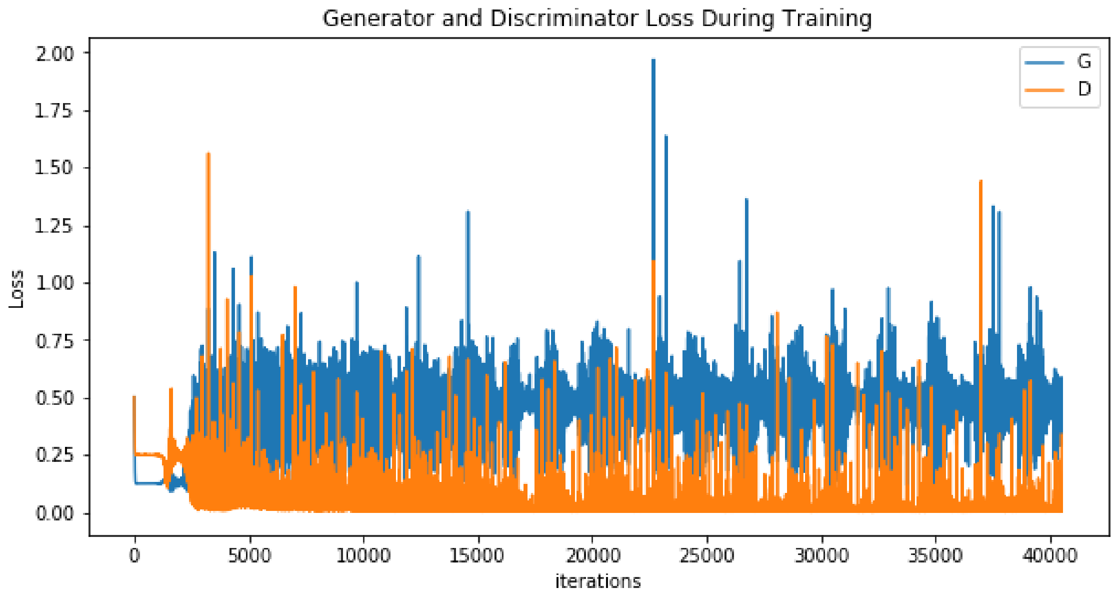
0.1 Results

Finally, lets check out how we did. Here, we will look at three different results. First, we will see how D and G's losses changed during training. Second, we will visualize G's output on the fixed_noise batch for every epoch. And third, we will look at a batch of real data next to a batch of fake data from G.

Loss versus training iteration

Below is a plot of D & G's losses versus training iterations.

[56]:
```
plt.figure(figsize=(10,5))
plt.title("Generator and Discriminator Loss During Training")
plt.plot(G_losses,label="G")
plt.plot(D_losses,label="D")
plt.xlabel("iterations")
plt.ylabel("Loss")
plt.legend()
plt.show()
```



Visualization of G's progression

Remember how we saved the generator's output on the fixed_noise batch after every epoch of training. Now, we can visualize the training progression of G with an animation. Press the play button to start the animation.

[57]:
```
print(type(img_list[0]))
print(img_list[0].shape)
for i in range(len(img_list)):
    img_list[i]=img_list[i].detach().cpu()
```

```
↪---------------------------------------------------------------------------

    IndexError                                Traceback (most recent call
↪last)

    <ipython-input-57-77092cc855f2> in <module>
 ----> 1 print(type(img_list[0]))
       2 print(img_list[0].shape)
       3 for i in range(len(img_list)):
       4     img_list[i]=img_list[i].detach().cpu()


    IndexError: list index out of range
```

[ ]:
```python
#%%capture
fig = plt.figure(figsize=(8,8))
plt.axis("off")
ims = [[plt.imshow(np.transpose(i,(1,2,0)), animated=True)] for i in img_list]
ani = animation.ArtistAnimation(fig, ims, interval=1000, repeat_delay=1000, blit=True)

HTML(ani.to_jshtml())
```

Real Images vs. Fake Images

Finally, lets take a look at some real images and fake images side by side.

[ ]:
```python
# Grab a batch of real images from the dataloader
real_batch = next(iter(dataloader))

# Plot the real images
plt.figure(figsize=(12,12))
plt.subplot(1,2,1)
plt.axis("off")
plt.title("Real Images")
plt.imshow(np.transpose(vutils.make_grid(real_batch[0].to(device)[:64], padding=5, normalize=True).cpu(),(1,2,0)))

# Plot the fake images from the last epoch
plt.subplot(1,2,2)
plt.axis("off")
plt.title("Fake Images")
plt.imshow(np.transpose(img_list[-1],(1,2,0)))
plt.show()
```

0.2 Where to Go Next

We have reached the end of our journey, but there are several places you could go from here. You could:

- Train for longer to see how good the results get
- Modify this model to take a different dataset and possibly change the size of the images and the model architecture
- Check out some other cool GAN projects here <https://github.com/nashory/gans-awesome-applicatio
- Create GANs that generate music <https://deepmind.com/blog/wavenet-generative-model-raw-audic

[ ]:
```python
print(len(img_list[0]))
#print(img_list[0])
print(img_list[0].shape)
```

#f=open('FakeImage.txt','w') from torchvision.utils import save_image

for idx, item in enumerate(img_list): save_image(item, 'FakeImageFolder2/img'+str(idx)+'.png')

#f.close()

```
[ ]:
```